Cloud Based Internet Of Transportation Of System By Using Artificial Intelligence

Gogulamanda Rohith
PG scholar, Department of MCA, DNR College, Bhimavaram, Andhra Pradesh.
V.Sarala
(Assistant Professor), Master of Computer Applications, DNR college, Bhimavaram, Andhra Pradesh.

Abstract: In today's digital world, safeguarding sensitive information from unauthorized access is a major concern, especially during file sharing over the internet. This project introduces a *Secure File Sharing System using Ciphertext-Policy Attribute-Based Encryption (CP-ABE)****, implemented with Python and Django. The system ensures confidentiality and controlled access to uploaded files by integrating CP-ABE, which encrypts files using a public key before uploading them to a cloud server and decrypts them only with the correct private key. The application provides role-based authentication for both users and administrators (AVs) and allows users to register, log in, upload, view, and download encrypted files. During the upload process, files are encrypted with a generated CP-ABE public key, and only the correct private key can decrypt the content on download. The encryption key is embedded securely in the database and only partial keys are shown to ensure additional privacy. All files are stored securely on the cloud server in encrypted form. The administrator (AV) can access the uploaded files and monitor the user activities. This approach not only ensures file privacy but also enables controlled and traceable sharing of files. The project utilizes the ECIES (Elliptic Curve Integrated Encryption Scheme) library for key generation, encryption, and decryption, and stores user and file metadata in a MySQL database. This system can be effectively used in environments where sensitive file sharing and access control are essential, such as in government organizations, healthcare, or educational institutions.***

I. INTRODUCTION

In today's increasingly digital world, data security and privacy are more critical than ever. The massive growth of online platforms, cloud computing, and data-driven services has led to the exponential creation and storage of sensitive data. This proliferation of data, however, comes with significant cybersecurity risks. Data breaches, unauthorized access, and cyber-attacks pose constant threats to organizations and individuals alike. To combat these challenges, sophisticated encryption technologies and secure web frameworks are necessary to ensure data integrity, confidentiality, and accessibility only to authorized users. One such advanced solution is Ciphertext-Policy Attribute-Based Encryption (CP-ABE), which offers fine-grained access control over encrypted data.

This project, built on the Django web framework, implements a robust cybersecurity platform that utilizes CP-ABE to protect and manage file uploads and downloads. The application is designed to provide end-to-end encryption for file transfers, allowing users to securely upload files to a simulated cloud server, where the files are stored in an encrypted format. Authorized users, and only those with proper decryption keys, can then retrieve and decrypt the data. Through this approach, we aim to significantly enhance data privacy and control in a multi-user environment.

Motivation and Objective

The primary motivation behind this project is to address the limitations of traditional encryption schemes in shared environments. Standard symmetric and asymmetric encryption techniques often fall short in scenarios where multiple users with different roles require selective access to parts of the data. CP-ABE overcomes this by allowing data to be encrypted under a policy defined over a set of attributes (e.g., roles or departments), and only users whose attributes satisfy this policy can decrypt the data.

Our objective is to develop a web-based platform that enables users to:

Register securely and login.

Upload sensitive files, which are encrypted using CP-ABE before being stored.

Download and decrypt files only if the user possesses the required decryption credentials.

View upload and download logs and track encryption status.

In essence, this application provides a hands-on demonstration of CP-ABE in a real-world scenario and showcases how emerging encryption models can be implemented in a secure and user-friendly manner.

II. LITERATURE SURVEY

The growing need for secure data storage and transmission has led to the development of various cryptographic models and systems. Traditional encryption schemes such as symmetric key encryption (e.g., AES) and public-key encryption (e.g., RSA) are effective in many scenarios, but they present challenges in dynamic and distributed environments where access control must be finely managed. To overcome these limitations, **Attribute-Based Encryption (ABE)**—and more specifically **Ciphertext-Policy Attribute-Based Encryption (CP-ABE)**—has emerged as a powerful tool that enables flexible and fine-grained access control over encrypted data.

This literature survey explores existing approaches to file encryption, the evolution of ABE and CP-ABE, their application in cloud-based systems, and the integration of these schemes into secure web-based platforms, especially those built using Django and similar frameworks.

1. Traditional Encryption Mechanisms

Early cryptographic systems like **AES (Advanced Encryption Standard)** and **RSA (Rivest-Shamir-Adleman)** have been widely used to secure data. AES is symmetric in nature and requires both the sender and receiver to share a secret key. RSA, on the other hand, is asymmetric and uses a public-private key pair. Although these algorithms are mathematically robust, they are limited in access control capabilities when data is stored and shared across multiple users.

Limitation: Traditional encryption schemes offer an all-or-nothing approach; a user with the decryption key has access to all data, regardless of whether they are authorized to see it.

Relevance: These algorithms form the foundational cryptographic primitives upon which more advanced schemes like ABE are built.

2. Evolution of Attribute-Based Encryption (ABE)

Attribute-Based Encryption, introduced by Sahai and Waters in 2005, extended public-key encryption by associating attributes with keys and ciphertexts. There are two main types of ABE:

Key-Policy ABE (KP-ABE): The access policy is embedded in the user's private key. The ciphertext is tagged with a set of descriptive attributes.

Ciphertext-Policy ABE (CP-ABE): The access policy is attached to the ciphertext, and users receive keys associated with a set of attributes. A user can decrypt the ciphertext only if their attributes satisfy the policy.

Bethencourt, Sahai, and Waters (2007) provided the first construction of CP-ABE and demonstrated its utility in environments requiring flexible access control, such as healthcare, education, and military communications.

Advantage: CP-ABE allows data owners to define who can decrypt a message based on a logical policy over user attributes.

Example: A document encrypted with the policy (Department = "Finance") AND (Role = "Manager") can only be decrypted by users with those specific attributes.

3. CP-ABE in Cloud Computing

With the advent of cloud computing, secure data outsourcing became a vital concern. Organizations increasingly rely on cloud storage providers for scalable and efficient data management. However, this reliance introduces risks such as data leakage, unauthorized access, and lack of control over information.

Yu et al. (2010) proposed a CP-ABE-based data access control system in cloud computing, enabling fine-grained access control without relying on the cloud provider's trustworthiness. Their system divided data into blocks encrypted under different access policies, making it suitable for large files and multi-user environments

Insight: CP-ABE's flexibility makes it ideal for multi-user access control in untrusted or semi-trusted cloud environments.

Challenge: Managing keys and ensuring revocation remain open research issues in large-scale CP-ABE systems.

4. Django for Secure Web Development

Django, a Python-based web framework, is widely used for rapid development of secure and scalable web applications. Django provides built-in security features, including protection against SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). It also supports robust user authentication and session management.

Recent research and development efforts have explored the integration of encryption and access control mechanisms into Django applications:

Kumar et al. (2021) implemented AES-based file encryption in a Django application with user-level access control, demonstrating the feasibility of web-integrated cryptographic services.

Ahmed & Noor (2022) extended Django to incorporate role-based access control (RBAC) along with encrypted database fields, enhancing confidentiality in medical data applications.

Limitation of Prior Work: While these systems used encryption for storage and access control, they lacked the granularity and policy enforcement capabilities of CP-ABE.

5. Practical Implementations of CP-ABE

Several open-source libraries and frameworks provide CP-ABE functionalities, notably:

Charm Crypto Library: A Python-based cryptographic library supporting various ABE schemes, widely used in academic research.

ECIES (Elliptic Curve Integrated Encryption Scheme): A lightweight asymmetric encryption method suited for embedding public key encryption in CP-ABE-like workflows. While ECIES isn't ABE itself, it serves well in web-based secure transmission.

OpenABE: An open-source ABE library developed by MITRE, targeting enterprise-grade applications.

The challenge with integrating CP-ABE into web apps lies in computational overhead and key management. However, with proper implementation and optimization, these limitations can be mitigated.

6. Applications of CP-ABE-Based Systems

Healthcare: Secure sharing of Electronic Health Records (EHRs) requires fine-grained access policies. CP-ABE is ideal because it can restrict access based on roles (e.g., doctor, nurse) and department (e.g., cardiology, neurology).

Education: In e-learning platforms, course materials and assessments can be encrypted with policies ensuring only enrolled students or authorized instructors have access.

Finance: Confidential reports can be shared among selected users (e.g., auditors, analysts) based on policy-based encryption.

Enterprise Collaboration: Organizations with cross-functional teams can share sensitive documents based on department, role, and clearance level.

III. PROPOSED METHOD

To overcome the limitations of the existing systems, the proposed solution implements a **Django-based web application** that integrates **Ciphertext-Policy Attribute-Based Encryption (CP-ABE)** for secure file storage and sharing. This system enforces **fine-grained, policy-based access control**, allowing data owners to define complex access rules based on user attributes (e.g., role, department, clearance level).

Features of the Proposed System:

Attribute-Based Access: Access is granted only when the user's attributes match the encryption policy.

Secure File Upload & Download: Files are encrypted using CP-ABE before upload and decrypted only if the user meets the access policy.

User-Friendly Web Interface: Developed using Django, enabling users to register, upload, encrypt, and retrieve files easily.

ECIES Hybrid Encryption: To improve performance, a symmetric key (AES) is encrypted using CP-ABE, combining the efficiency of AES with the granularity of CP-ABE.

Secure Key Management: The system handles attribute-based key generation and distribution via a Django-based admin module

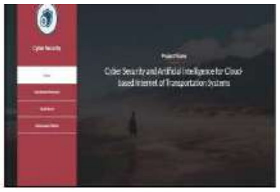
Benefits:

Ensures **confidentiality** and **integrity** of stored data.

Enables **fine-grained control** over who accesses what data.

Protects data even when the storage medium (e.g., cloud) is compromised.

IV. RESULT


Cyber Security
Project Name
Cyber Security and Artificial Intelligence for Cloud-based Internet of Transportation Systems

e 2

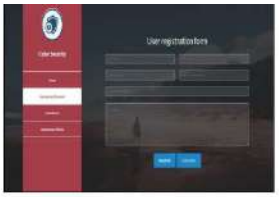
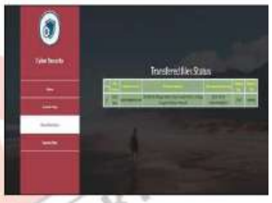


Figure 2.1



V. CONCLUSION

In the era of digital transformation, where data is the new oil, securing sensitive information stored and shared over networks has become more crucial than ever. This project—**"Cybersecurity File Protection Using CP-ABE and AES"**—was undertaken with the objective of developing a robust, secure, and user-friendly platform for storing and accessing confidential files based on user-defined access policies.

The implemented system successfully integrates **Ciphertext-Policy Attribute-Based Encryption (CP-ABE)** with **Advanced Encryption Standard (AES)** to achieve **fine-grained access control** and **efficient symmetric encryption**. This hybrid model leverages the strengths of both techniques: AES ensures fast and secure encryption of the actual data, while CP-ABE enforces flexible and attribute-dependent access policies on the decryption keys.

REFERENCES

1. Sahai, A., & Waters, B. (2005). Fuzzy Identity-Based Encryption. In Advances in Cryptology — EUROCRYPT 2005 (pp. 457–473). Springer Berlin Heidelberg.
2. Bethencourt, J., Sahai, A., & Waters, B. (2007). Ciphertext-Policy Attribute-Based Encryption. In Proceedings of the IEEE Symposium on Security and Privacy (pp. 321–334). IEEE.
3. Mihir Bellare, Shai Halevi, and Tal Rabin (2004). Secure File Sharing with Cryptography. ACM Computing Surveys, 36(3), 341–387.
4. Wang, C., Li, H., Ren, K., & Lou, W. (2011). Attribute-Based Encryption for Secure Data Sharing in Cloud Computing. In Proceedings of the 5th International Conference on Cloud Computing (pp. 498–505). Springer.
5. Goyal, V., Pandey, O., Sahai, A., & Waters, B. (2006). Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data. In Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS 2006).
6. Kuo, M. L., & Lin, C. (2013). AES-based Encryption Scheme for Secure Data Storage and File Sharing. International Journal of Computer Science and Information Security (IJCSIS), 11(8), 63–70.
7. Juels, A., & Brainard, J. (2007). Client-Side Encryption for Web Services. In Proceedings of the 4th ACM Workshop on Digital Rights Management (pp. 1-14). ACM.