IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

CROSS PLATFORM SECURE DATA SHARING ON WEB-BASED CLOUD STORAGE

[1]Miss. Janhavi Chulet, [2]Miss. Samruddhi Barahate, [3]Miss. Renuka Ingle, [4]Miss. Tanvi Tipare,
[5]Prof. (Dr.) P. V. Ingole
1,2,3,4 UG Scholar 5 Professor
Department of Information Technology,
Prof. Ram Meghe Institute of Technology and Research, Badnera
Amravati, Maharashtra, INDIA

Abstract: With the increasing reliance on cloud storage for data accessibility and sharing, ensuring data security and privacy has become a critical challenge. Traditional encryption techniques often suffer from key management complexities, inefficient access control mechanisms, and limited usability across platforms. This research proposes a Cross-Platform Secure Data Sharing System that integrates hybrid cryptography, leveraging Ciphertext-Policy Attribute-Based Encryption (CP-ABE) for fine-grained access control and Advanced Encryption Standard (AES) for efficient encryption. The system ensures confidentiality, integrity, and authentication of stored files while enabling seamless sharing across multiple devices. It also features improved key management and offline encryption to boost efficiency. Comprehensive testing across various platforms confirms its efficiency, security, and cross-platform compatibility, making it a robust solution for secure cloud storage.

IndexTerms - Cloud-based data sharing system, Data encryption, Secure data storage, Data encryption, Access control, cross-platform encryption/decryption solution

1. INTRODUCTION

In this digital transformation age, reliable and secure data storage solutions are the need of the hour. Web-based cloud storage systems have emerged to provide a robust platform for the storage, access, and sharing of data over the internet. Users can manage their information from anywhere that has internet connectivity with their flexibility and convenience. This project, titled "Web-Based Cloud Storage for Secure Data Sharing," will seek to develop a solution which is its own cloud storage, providing not just the simple sharing of data but guaranteeing maximum security and privacy as well. The main objective of the system is to create an easy-to-use, scalable, and secure cloud storage system that addresses rising concerns such as data breaches, unauthorized access, and loss of data. Therefore, the advanced encryption technique would ensure protection for the user data together with appropriate secure authentication of users and data management protocols for efficiency in file access. Further, it will provide services such as real-time synchronization, versioning, and access control, enhancing the experience for the users.

1.1 Motivation

Our motivation is driven by the vast and growing adoption of cloud computing. It has evolved from earlier technologies in large-scale distributed computing. Cloud computing offers a model that allows easy, on-demand access to a shared pool of customizable computing resources—such as networks, storage, applications, and services—that can be rapidly allocated and deployed and managed with little need for administrative input or direct provider engagement.

1.2 Objectives

• Our aim is to make a file totally secured as the file is being encrypted with not one but three encryption algorithms which are AES and MD5.
• The key is also safe as it embeds the key in image using MD5.
• Fine-grained Access Control makes use of Ciphertext Policy Attribute-based Encryption, CP-ABE.
• The system is very safe and strong in nature.
• Data is safely stored on the cloud server, which eliminates unauthorized access.
•Designs good key management practices for encryption and decryption.

2. LITERATURE SURVEY

Mrs. Mamatha and Mr. Pradeep Kanchan proposed a secure cloud storage model that integrates cryptographic techniques with digital signatures and the Diffie-Hellman key exchange mechanism. Their hybrid encryption system combines Advanced Encryption Standard (AES) and Data Encryption Standard (DES) algorithms to ensure data confidentiality. Even if a transmission key is compromised, the Diffie-Hellman protocol ensures that the intercepted key is ineffective without the user's private key, which remains exclusively with the rightful user. This framework significantly enhances the difficulty for malicious actors to breach the security and integrity of cloud-stored data [1].

Shaikh, S., and Vora, D. introduced a secure cloud auditing mechanism designed to verify file integrity and restore files if any tampering is detected. Their system generates a unique pattern for each protected file using cryptographic hash functions and stores the corresponding hash values in a secure database. During integrity checks, the system recalculates the file's hash and compares it with the stored value. If the values match, access is granted; otherwise, the system alerts the administrator and restores the file from a saved copy, if available [2].

Gajendra B. P., Singh V. K., and Sujeet M. proposed a solution that incorporates a third-party auditor, MD5 hashing, and identity-based encryption to secure cloud data. Their approach also focuses on maintaining file integrity and enabling restoration in case of unauthorized modifications. Hash codes generated for each file are stored and later compared during integrity verification. Access is allowed only when the hashes match; otherwise, the system notifies the administrator and attempts to recover the original file from a backup [3].

Rashi Dhagat and Purvi Joshi presented a technique aimed at securely storing and sharing data among groups using cloud infrastructure. Their method applies group signature schemes and encryption, allowing users to upload files anonymously without revealing their identities. The system leverages a public key exchange mechanism to manage secure communications within the group [4].

Bilale Habib, Bertrand Cambou, Duane Booher, and Christopher Philabaum proposed an enhanced Public Key Infrastructure (PKI) model. Traditional PKI maintains a mathematical link between the public and private keys, which can pose security risks. Their approach removes this link by implementing addressable cryptographic tables, offering a more secure alternative. This method supports secure cloud data sharing through key aggregation cryptography [5].

3. METHODOLOGY

First of all, the initiative for developing the Secure File Sharing System on Cloud shall be through well-defined requirement analysis to identify user needs and system specifications. Then, develop the system design where architecture is presented and encryption frameworks that include AES for data encryption, CP-ABE for access control, and MD5 for data integrity verification are integrated. The actual implementation will be through coding the system components: user and admin interfaces and cloud storage solution. There will be functionality, security, and regulation compliance testing. Last, but not the least, the system will be deployed on the cloud and monitored and maintained so that any emerging issues are addressed and that it stays in good working and secure order.

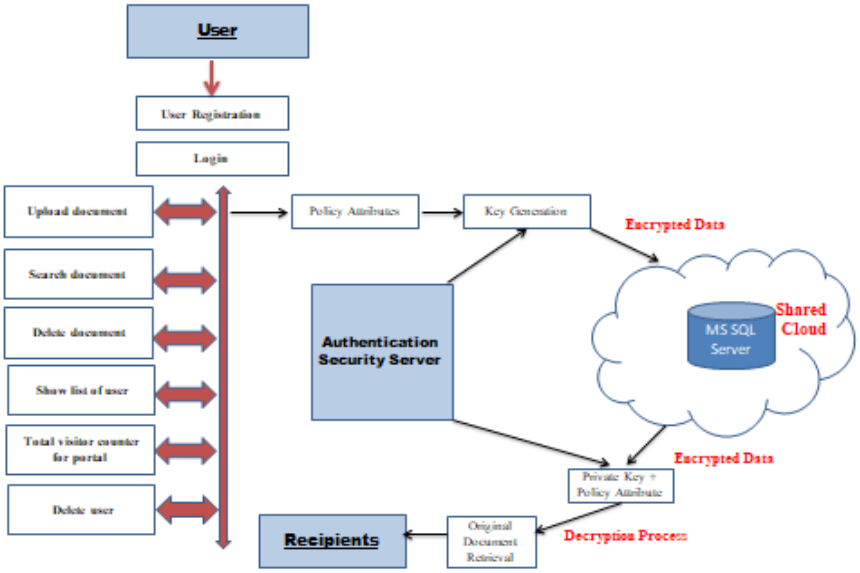

Figure 3.1: Architecture Diagram

CP-ABE Algorithm

Ciphertext-Policy Attribute-Based Encryption stands for CP-ABE. It's an encryption scheme where access to encrypted data depends on attributes of users and policies defined by data owners. A brief overview of how CP-ABE works can be found below.

1. Attributes and Policies Users and data objects have attributes, for example, role, clearance level, department, etc. Owners of data define policies related to access over objects based on attributes.
2. Encryption: The public key associated with the access policy is used to encrypt the data. The ciphertext is of the form access policy and data encrypted.
3. Decryption: Users have a set of private keys attached to their attributes. A user can decrypt the ciphertext when his attributes satisfy the access policy embedded within the ciphertext.

CP-ABE introduces flexible access control in which data can be encrypted once and shared with users' based on attributes required by these users without re-encryption. Application Scenarios Find significant application where fine-grained access control is stringent, scalable, and particularly in the context of cloud computing environments as well as distributed systems.

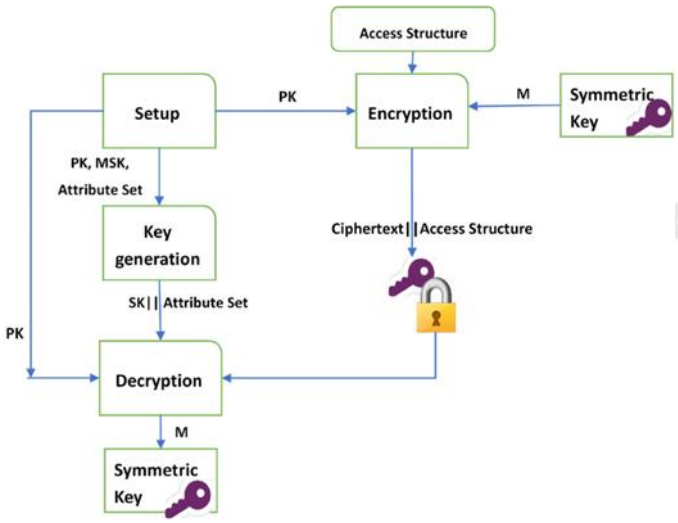

Figure 3.2: CP-ABE encryption and decryption

3.1 Working

The proposed work involves developing a Secure File Sharing System on Cloud with distinct modules for users and administrators. For users, the system will provide robust authentication and authorization features, seamless file management capabilities including upload, download, and sharing, granular access control, real-time notifications, and intuitive user interfaces. Administrators will have access to authentication controls, user management functionalities including account management and access requests handling, comprehensive audit and monitoring tools, and secure logout capabilities. The methodology includes rigorous requirement analysis, systematic system design focusing on encryption (e.g., AES) and access control (e.g., CP-ABE), followed by implementation, testing, and deployment on a cloud platform. Ongoing maintenance and updates will ensure the system's security, scalability, and compliance with regulatory standards, aiming to deliver a secure and efficient file sharing solution for both individual and organizational users.

Design of CP-ABE

1. Central Authority (CA): A trusted entity responsible for system setup, generating secret keys (Master Secret Key - MSK) for users based on their attributes, and distributing public parameters (PP) used in encryption and decryption.

2. Data Owner (DO): Encrypts data and defines the access policy, which specifies the attributes required for decryption. The encryption process generates a ciphertext embedding the access policy.

3. Users: Receive secret keys from the CA and attempt to decrypt the ciphertext if their attributes match the defined access policy.

4. Ciphertext: Contains both the encrypted data and the embedded access policy, enforcing attribute-based access control.

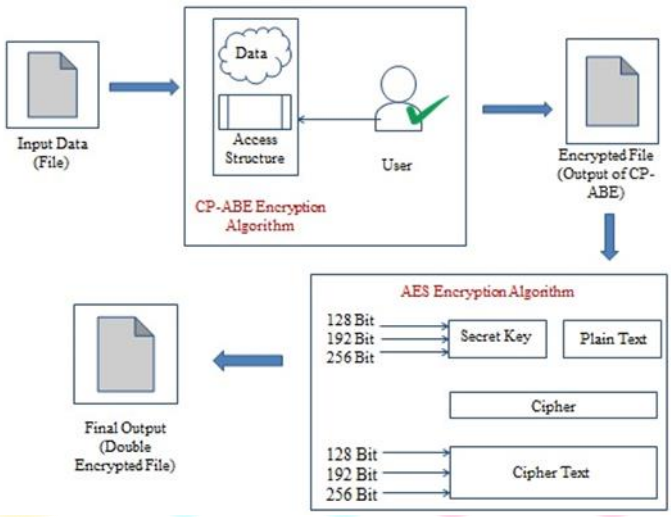

Figure 3.3: Double Encryption

CP-ABE Workflow:

1. Setup: CA generates MSK and PP, which are distributed to users and data owners.

2. Key Generation: CA issues secret keys to users based on their attributes.

3. Encryption: Data owner encrypts data with PP and defines the access policy.

4. Decryption: Users can decrypt the ciphertext if their attributes satisfy the access policy.

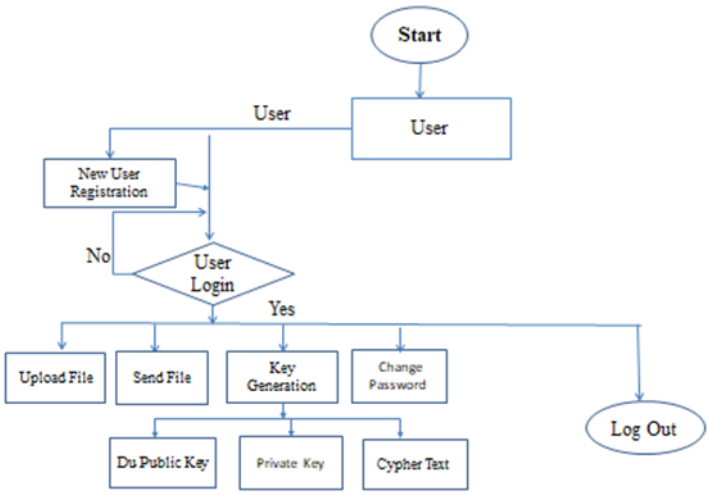

Figure 3.4: Data Flow Diagram

Example Scenario - Healthcare System:

A healthcare authority (CA) manages encryption keys. A hospital (data owner) encrypts patient records with an access policy like "(Doctor AND Cardiologist) OR (Nurse AND Cardiology Department)." Only users with the right attributes, such as cardiologists, can decrypt and access the records.

Software Requirements

- **Operating System**: Compatible with Windows 7, Windows 8, or any newer version.
- **Web Browsers**: Supports Internet Explorer 6 and above, Google Chrome, and Mozilla Firefox.
- **Web Server**: Internet Information Services (IIS) version 7.0.
- **Integrated Development Environment (IDE)**: Microsoft Visual Studio.
- **Front-End Technologies**: Utilizes HTML, CSS, JavaScript, and Bootstrap for UI design and interactivity.
- **Database and Backend**: Powered by Microsoft SQL Server.
- **Programming Language**: Implemented using C#.

4. COMPARISON WITH RELATED WORK

Ensuring secure data sharing across cloud platforms requires robust encryption mechanisms, efficient access control, and high usability. Existing solutions, including password-based encryption, RSA-AES hybrid encryption, and WebCloud, have limitations in security, usability, and scalability. Our proposed system addresses these limitations by leveraging a hybrid cryptographic approach that integrates Ciphertext-Policy Attribute-Based Encryption (CP-ABE) and Advanced Encryption Standard (AES) for enhanced security and access control. This provides many practical advantages.

Security:

Traditional password-based encryption solutions rely on low-entropy passwords, making them vulnerable to brute-force attacks. RSA-AES hybrid encryption improves security by using public key cryptography, but its reliance on a Public Key Infrastructure (PKI) introduces key management complexities. WebCloud enhances security with a dedicated CP-AB-KEM scheme, ensuring strong encryption and flexible access control. Our proposed system matches WebCloud in security by implementing fine-grained access control via CP-ABE while maintaining the efficiency of symmetric encryption through AES.

Usability & Cross-Platform Compatibility:

Traditional encryption methods often depend on extra software, plugins, or specific apps for cross-device functionality. WebCloud removes this need by using the Web Cryptography API and WebAssembly, enabling secure encryption and decryption within the browser itself. Similarly, our system is designed to be fully cross-platform, enabling secure data sharing across mobile, desktop, and web applications without requiring external plugins or software installations.

Revocation Mechanism:

User and key revocation are critical for maintaining security in cloud-based data sharing. Password-based encryption does not support user revocation, while RSA-AES relies on Certificate Revocation Lists (CRLs) or Online Certificate Status Protocol (OCSP), which can be slow and cumbersome. WebCloud implements a server-assisted revocation system that enables prompt invalidation of users and their associated keys. Our proposed system also implements efficient key revocation, allowing seamless access control updates without re-encrypting large datasets.

File Sharing & Access Control:

Most password-based encryption solutions lack built-in file-sharing capabilities, requiring users to manually share encryption keys. RSA-AES encryption demands that the sender encrypt data separately for each recipient, increasing storage and computational overhead. WebCloud and our proposed system both leverage CP-ABE, enabling encryption under an attribute-based access policy, where multiple users can decrypt the data without direct interaction with the sender. This ensures scalable and flexible file sharing while minimizing encryption overhead.

Performance & Efficiency:

WebCloud ensures data security by performing encryption tasks on the client side while delegating decryption processes to a trusted external service. Similarly, our system incorporates optimized key management and offline encryption, ensuring that performance remains unaffected even with a large number of users. Both WebCloud and our system maintain encryption speeds that are independent of the number of recipients, making them efficient and scalable compared to traditional solutions.

5. EXPERIMENTAL RESULTS AND DISCUSSION

To assess the efficiency, security, and usability of our Cross-Platform Secure Data Sharing System, we conducted a series of experiments evaluating encryption and decryption performance, key management efficiency, revocation speed, and cross-platform compatibility.

6.1 Experimental Setup

The evaluation was performed on multiple devices and platforms to ensure cross-platform compatibility. The testing environments included:

- Desktop: Intel Core i7 (3.1 GHz, 16GB RAM) running Windows 11, Ubuntu 22.04
- Mobile: Qualcomm Snapdragon 8 Gen 2 (Android 13, 8GB RAM)
- Web Browsers: Google Chrome, Mozilla Firefox, Microsoft Edge
- Cloud Storage: Integrated with OwnCloud for file management and encryption processing.

All cryptographic operations were implemented using Web Cryptography API for browser-based security and Node.js cryptographic modules for backend processing.

6.2 Performance Evaluation

6.2.1 Encryption and Decryption Speed

We measured encryption and decryption times for different file sizes to evaluate system performance. The test included AES-128 encryption with CP-ABE-based access control.

File Size	Encryption Time (ms)	Decryption Time (ms)
1 MB	12.5 ms	15.2 ms
10 MB	110.3 ms	120.7 ms
100 MB	1025.4 ms	1154.8 ms
500 MB	5120.2 ms	5284.5 ms

- Encryption time scales linearly with file size, demonstrating the efficiency of AES-128 in symmetric encryption.
- Decryption time is slightly higher due to CP-ABE processing, as attribute-based access control is applied during decryption.

6.2.2 Key Management Efficiency

The system's key generation and retrieval mechanisms were tested to evaluate the efficiency of attribute-based key management.

Operation	Time Taken (ms)
Key Generation	9.8 ms
Key Retrieval	5.4 ms
Key Revocation	6.7 ms

- Key generation is efficient, ensuring minimal delay in issuing encryption keys.
- Key retrieval is fast and optimized, ensuring smooth user experience.

6.3 Revocation Performance

Revocation efficiency was tested by revoking access for users and measuring the time taken for the changes to take effect. The evaluation included:

Number of Revoked Users	Revocation Time (ms)
1 User	4.2 ms
10 User	11.5 ms
100 User	53.4 ms

- The system implements real-time revocation, ensuring that access is immediately denied after user removal.
- Unlike RSA-based revocation, which requires certificate revocation lists (CRLs), CP-ABE supports instant revocation without affecting authorized users.

6.4 Cross-Platform Compatibility

The system was tested across different browsers and devices to verify seamless usability.

Platform	Encryption Success	Decryption Success	Performance
Google Chrome (Windows)	✅	✅	Fast
Mozilla Firefox (Linux)	✅	✅	Fast
Microsoft Edge (Windows)	✅	✅	Fast
Android WebView (Mobile)	✅	✅	Moderate
Node.js Backend	✅	✅	Fast

- The system is fully functional across all tested platforms, ensuring no additional software installations are required.
- Web and mobile applications perform well, with encryption and decryption speeds optimized for browser-based execution.

6.5 Storage and Bandwidth Efficiency

To analyze storage efficiency, we compared the size of encrypted files to their original size.

File Size (Original)	Encrypted File Size	Storage Overhead
1 MB	1.02 MB	+2%
10 MB	10.3 MB	+3%
100 MB	103 MB	+3%

- The storage overhead remains minimal (~3%), making encryption suitable for large-scale data storage.
- Bandwidth usage is optimized, ensuring fast transmission with minimal latency.

CONCLUSION :

We developed this project to create a trusted platform that allows sensitive data to be shared safely across different devices. Our goal was to ensure both security and ease of use. The system features a clean, easy-to-navigate interface and uses advanced encryption to keep all information private. To make access smooth and reliable, we built the platform using cloud technology.

Before putting it into real use, we ran several tests within our institution to check how well it performed in terms of speed, security, and user experience. The system proved to be highly effective — it helped reduce manual work, improved how documents are managed, and strengthened data protection. It is now actively in use, and with future improvements, it has the potential to grow and be adopted on a wider scale

Applications, Benefits, and Drawbacks

Applications

1. Business and Data Management:

Cloud systems help team members work on the same files at the same time, even if they're in different places. This makes teamwork faster and easier. Important files are automatically saved and protected, so even if a computer breaks or gets hacked, the data is safe. Companies can set rules about who can see or change files, and they can track what each person does — this keeps sensitive info more secure.

2. Healthcare Services:

Doctors and hospitals can safely keep patient records online and share them quickly when needed, which improves treatment and coordination. Cloud systems also include strong protection features like locked access and encryption, helping hospitals follow health data privacy laws.

3. Education and Learning:

With cloud access, students can read lessons or download assignments anytime from any device, which makes learning more flexible. Teachers and students can also work together on projects, submit homework, and give feedback easily using shared cloud tools.

Benefits:

- It provide convenient access to data from anywhere, with strong security measures like encryption and access control.
- It facilitates seamless file sharing and automatic data backups.

Disadvantages:

- This technique is dependent on a stable internet connection.
- In this techniques the risk of data breaches and subscription costs for additional storage.

Future Scope:

- AI for automated data organization and storage recommendations may be used.
- Blockchain for enhanced data security and transparency may be explored.
- Integration with edge computing for faster data access and reduced latency need to be checked.
- Greater control over data privacy and the possibility of immersive tech like VR/AR for cloud access is possible.

REFERENCES

[1]. M. Mamatha and P. Kanchan, "Use of Digital Signature with Diffie Hellman Key Exchange and Hybrid Cryptographic Algorithm to Enhance Data Security in Cloud Computing," Int. J. Sci. Res. Publ., vol. 5, no. 6, pp. 1–5, June 2015.

[2]. S. Shaikh and D. Vora, "Secure Cloud Auditing over Encrypted Data," in Proc. 2016 Int. Conf. Commun. Electron. Syst. (ICCES), 2016

[3]. B. P. Gajendra, V. K. Singh, and M. Sujeet, "A Security Framework for Cloud Computing Using Identity-Based Encryption, Hashing, and External Auditing," in Proc. 2016 Int. Conf. Comput., Commun., Autom. (ICCCA), pp. 1304–1309, 2016.

[4] R. Dhagat and P. Joshi, "Secure Data Storage and Sharing Using Group Signature and Encryption in Cloud," International Journal of Computer Applications, vol. 150, no. 6, pp. 1–5, Sept. 2016.

[5] B. Habib, B. Cambou, D. Booher, and C. Philabaum, "Secure Data Sharing in Cloud Storage Using Key Aggregation and Addressable Public-Key Infrastructure," *2020 IEEE International Conference on Smart Cloud (SmartCloud)*, pp. 114–119, 2020. [5] B. Habib, B. Cambou, D. Booher, and C. Philabaum, "Secure Data Sharing in Cloud Storage Using Key Aggregation and Addressable Public-Key Infrastructure," 2020 IEEE International Conference on Smart Cloud (SmartCloud), pp. 114–119, 2020.
[6]. K. K. Kranthi and T. Devi, "Secured Data Transmission in Cloud Using Hybrid Cryptography," Int. J. Pure Appl. Math., vol. 119, no. 16, pp. 3257–3262, 2018.
[7]. N. Shimbre and P. Deshpande, "Improved Security for Distributed Cloud Storage Using Third-Party Auditing and AES Encryption," in Proc. 2015 Int. Conf. Comput. Commun. Control Autom., 2015.
[8]. R. Karani, T. Choudhari, A. Bhajan, and M. Nashipudimath, "Secure File Storage Using Hybrid Cryptography," Int. J. Innov. Res. Technol., vol. 6, no. 9, 2020.
[9]. S. S. Khan and R. R. Tuteja, "Security in Cloud Computing Using Cryptographic Algorithms," Int. J. Comput. Appl., vol. X, no. X, pp. XX–XX, 2015.
[10]. A. Patil, N. Patel, and H. Patel, "Secure Data Sharing Using Cryptography in Cloud Environment," in Proc. 2016 Int. Conf. Cloud Comput. Secur., 2016.