Pinnacle Academic Press
Proceedings Series

Vol. 2 2025

Article Open Access

Research on Cloud Storage Data Access Control Based on the CP-ABE Algorithm

Yu Pan [1,*]

[1] Matthews Real Estate Investment Services, El Segundo, California, 90245, USA
* Correspondence: Yu Pan, Matthews Real Estate Investment Services, El Segundo, California, 90245, USA

Received: 02 April 2025
Revised: 13 April 2025
Accepted: 10 May 2025
Published: 12 June 2025

Abstract: With the rapid development of cloud computing technology, cloud-based data storage has become a common choice for both enterprises and individual users. However, data security and access control in cloud storage environments have emerged as critical challenges. This paper focuses on attribute-based encryption algorithms for access control (CP-ABE) and proposes an improved CP-ABE model tailored for cloud storage access control needs. This model employs flexible attribute policies to achieve precise control over data access in the cloud, enhancing both security and privacy. The paper begins by introducing the basic principles and improvements of the CP-ABE algorithm, followed by the design and implementation of a data access control system. A detailed analysis of the system's performance and security is provided. Experimental results indicate that the proposed model demonstrates significant advantages in terms of efficiency and security in data access control. This research provides a potentially effective technical approach for secure data access in cloud storage environments.

Keywords: cloud storage; data access control; Ciphertext-Policy Attribute-Based Encryption (CP-ABE); data security; privacy protection

1. Introduction

With the rapid advancement of information technology, cloud computing has become a vital means for data storage and management, providing convenient, efficient, and flexible storage solutions for both individual users and enterprises. However, the widespread use of cloud storage poses major challenges to data security and privacy protection. In traditional data storage environments, users typically have substantial control over data access permissions. However, the centralized nature of cloud storage means that data security is largely dependent on the security measures of cloud service providers. In the event of data breaches or unauthorized access on the cloud platform, users' sensitive information faces considerable risk. Thus, achieving efficient and secure access control in cloud storage environments has become a primary focus and challenge for research. Attribute-Based Encryption (ABE) technology, which enables encryption based on identity attributes, provides a novel approach to access control in cloud storage. Particularly, the Ciphertext-Policy Attribute-Based Encryption (CP-ABE) model combines access control with user attributes through encryption policies, making access control more flexible and fine-grained. CP-ABE allows data owners to set access policies during encryption, determining whether a user can decrypt data based on their attribute set. This approach ensures data security while reducing the administrative burden of managing access permissions in cloud storage environments. However, traditional CP-ABE algorithms face challenges in practical applications, such as high computational overhead, complex key management, and difficulty in dynamically updating attributes. These limitations affect the efficacy of CP-ABE in cloud storage settings, especially in scenarios with high-frequency data access and real-time requirements. To address these issues, this paper presents an improved CP-ABE model and designs a data access control system based on this model. The system enhances the efficiency and security of data access control through optimized encryption policies and key distribution mechanisms. This research aims to provide an effective solution for data access control in cloud storage environments. The feasibility and advantages of the improved model are verified through experiments. The structure of this paper is as follows: Section 2 reviews related work on data access control in cloud storage and the application of CP-ABE; Section 3 details the principles and improvements of the CP-ABE algorithm; Section 4 introduces the design of a data access control system based on the improved CP-ABE model; Section 5 presents system performance tests and result analysis; Section 6 provides a security and privacy analysis of the model. Finally, the paper concludes with a summary of the research and prospects for future work [1].

2. Overview of CP-ABE Algorithm for Cloud Storage Data Access Control

In cloud storage environments, data security and privacy protection are critical concerns, especially in multi-user settings where different users often have varying levels of data access permissions. Traditional access control methods generally rely on centralized servers to manage user permissions, which may result in potential single points of failure and increased complexity in permission management. Attribute-Based Encryption (ABE) offers a decentralized approach to access control by associating access permissions with user attributes, enabling fine-grained control. Ciphertext-Policy Attribute-Based Encryption (CP-ABE), a variant of ABE, allows data owners to define access policies during encryption. Only users who meet specific attribute requirements are able to decrypt the data. Consequently, CP-ABE has been widely applied in cloud storage systems as a more flexible and efficient solution for data access control. The core concept of CP-ABE is to define data encryption policies using an "attribute tree", as shown in Figure 1. Figure 1 illustrates a typical CP-ABE access control tree structure, comprising logical operation nodes (such as "AND" and "OR") and specific attribute nodes (e.g., "TWG", "SE", "RC"). This tree structure enables flexible permission management by breaking down complex access control rules into hierarchical conditions, allowing different users to decrypt data based on their attributes [2].

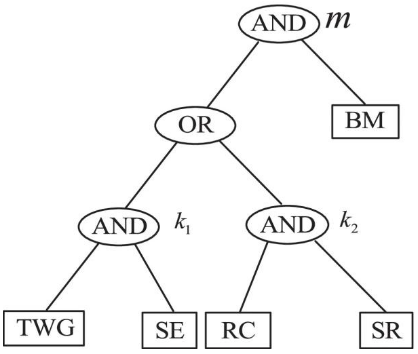

Figure 1. CP-ABE Attribute Structure Tree for Cloud Storage Data Access Control.

In Figure 1, the root node is an "AND" operation, indicating that the user must satisfy both child nodes' conditions to decrypt the data. One child node is the attribute "BM", representing a specific attribute requirement, while the other child node is an "OR" operation that contains two "AND" branches. This "OR" node allows users to decrypt data by satisfying either "AND" branch, with each "AND" branch containing more specific attribute requirements. For example, the left "AND" branch includes the attributes "TWG" and

"SE", while the right "AND" branch includes "RC" and "SR". This structure enables users to gain access if their attribute set satisfies the conditions of a complete branch, thus granting them decryption rights. The access control structure in CP-ABE is particularly suited for multi-level permission management in cloud storage environments. Different user roles and identities can automatically match the corresponding access permissions based on their attribute combinations without relying on a centralized server for real-time permission determination [3]. This decentralized control method not only improves system reliability but also significantly reduces administrative complexity. Moreover, CP-ABE supports dynamic attribute updates, meaning that when a user's permissions or attributes change, key re-distribution can adjust their access rights. This dynamic capability provides CP-ABE with a notable advantage in addressing complex and frequently changing access control requirements. In summary, the application of CP-ABE in cloud storage greatly enhances the flexibility and security of data access control. Through its attribute tree structure, CP-ABE enables multi-level, fine-grained access management, overcoming the limitations of traditional centralized access control. Subsequent sections of this paper will further explore the specific implementation details of the CP-ABE algorithm and verify its advantages in performance and security through experimental analysis [4].

3. Cloud Storage Data Access Control Model and Architecture Design Based on the CP-ABE Algorithm

3.1. System Architecture Design

The CP-ABE-based cloud storage data access control system integrates key roles to ensure secure and flexible access control. Figure 2 illustrates the interactions and data flows among the Data Owner, User, Proxy Server, Third Party, Attribute Authorities (AAs), and Trusted Entity, which enable access control in cloud storage [5].

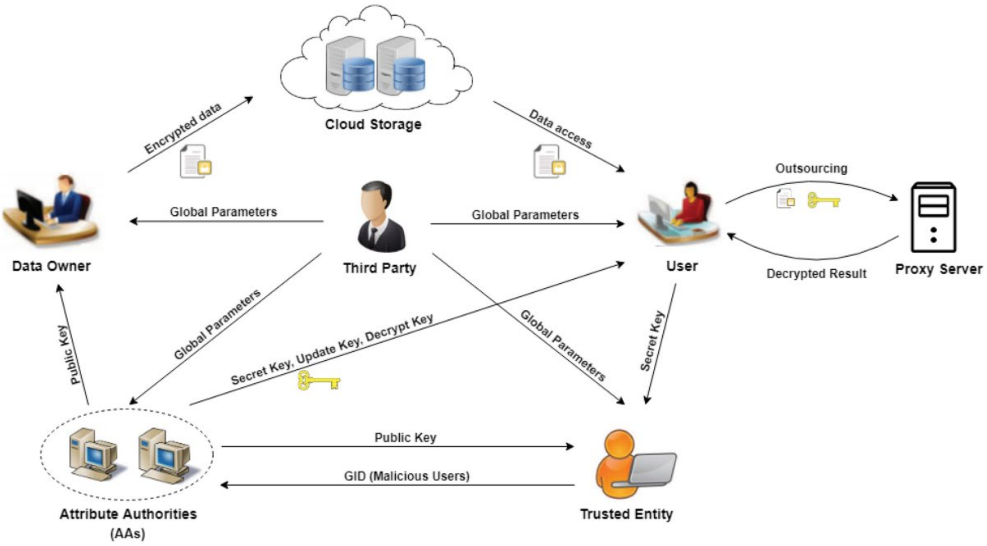

Figure 2. Cloud Storage Data Access Control System Architecture Based on CP-ABE.

The Data Owner encrypts data using the global parameters and public keys provided by the Attribute Authority, setting an attribute-based policy that allows only users with a specific attribute set to decrypt the data. After encryption, the data is stored in the cloud. To access data, a User must obtain a decryption key from the Attribute Authority, tailored to their attributes and aligned with the Data Owner's encryption policy. While Users can directly decrypt data, they can offload part of the decryption process to a Proxy Server, reducing computational load. The Proxy Server performs an initial decryption step using the User's key and partial data, generating an intermediate result that the User can further decrypt. This approach allows Users with limited resources to access encrypted data with minimal privacy risks, as the Proxy Server does not fully decrypt or directly access the complete data. The Proxy Server plays a vital role by handling outsourced decryption requests, easing Users' computational demands while preserving data security. Additionally, the Attribute Authority (AA) manages access control by generating and distributing decryption keys based on each User's attributes, enabling dynamic authorization. The AA's key distribution mechanism supports fine-grained access control, offering flexible permission management across different roles [6]. A Third Party is also included to generate and publish global system parameters, which are critical for encryption and decryption but do not participate in the data access or decryption processes. This third-party role ensures system uniformity and scalability, allowing Data Owners and Users to access public parameters without direct interaction. To counteract malicious activity, a Trusted Entity monitors and manages Global Identifiers (GIDs) for malicious users, sharing any detected GIDs with the Attribute Authority and other roles, effectively blocking unauthorized access. This feature strengthens security and defends against external threats. In conclusion, this architecture achieves secure, efficient, and flexible data access control through the combined efforts of all roles, addressing the complex requirements of cloud storage access control [7].

3.2. CP-ABE Algorithm Model for Cloud Storage Data Access Control

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is a flexible algorithm that enforces attribute-based access control through ciphertext access policies. The core of the CP-ABE algorithm model lies in defining access policies and performing encryption and decryption based on the attributes possessed by users. The following describes the encryption, key generation, and decryption processes in detail, along with related formulas for ease of calculation and implementation. During system initialization, the Attribute Authority (AA) generates the public parameters and the master key. The public parameters are used to encrypt data, while the master key is used to generate user private keys. First, a bilinear group G with generator g is chosen, and a hash function is defined as $\mathrm{H}: \{0,1\} * \rightarrow \mathrm{G}$. The master key MK and public parameters PK are generated as shown in Formula 1,2:

$$\mathrm{MK} = (\mathrm{g}, \mathrm{g}^{\alpha}) \tag{1}$$

$$\mathrm{PK} = (\mathrm{G}, \mathrm{g}, \mathrm{e}(\mathrm{g}, \mathrm{g})^{\alpha}) \tag{2}$$

Where α is a randomly selected private parameter, G is a cyclic group with bilinearity, g is the generator of G, and $\mathrm{e}: \mathrm{G} \times \mathrm{G} \rightarrow \mathrm{GT}$ is a bilinear mapping function. The public parameters PK and master key MK will be used in the subsequent steps for encryption and decryption operations. In the encryption phase, the Data Owner defines an access policy T and encrypts the message M using the public parameters PK. The access policy is usually represented by an access tree, where each leaf node is an attribute [8]. Assuming the message to be encrypted is M with access policy T, the ciphertext CT is generated as follows: A random number $\mathrm{s} \in \mathrm{Z}_{\mathrm{p}}$ is chosen, and the ciphertext CT is computed as shown in Formula 3:

$$\mathrm{CT} = (\mathrm{C} = \mathrm{M} \cdot \mathrm{e}(\mathrm{g}, \mathrm{g})^{\alpha \cdot \mathrm{s}}, \mathrm{C}' = \mathrm{g}^{\mathrm{s}}, \{\mathrm{Cx} = \mathrm{g}^{\mathrm{qx}(0)}, \mathrm{D}x = \mathrm{H}(\mathrm{attr}(\mathrm{x}))^{\mathrm{qx}(0)}\}\mathrm{x} \in \mathrm{T} \tag{3}$$

Where qx is a polynomial defined at each node x, C' is a part of the ciphertext related to the shared key *s*, and the constant term of the polynomial at the root node is *s*. Dx is the encrypted part of the attribute, which, combined with the output of hash function *H*, represents the encryption condition for specific attributes, used for attribute matching. This ciphertext structure ensures that only users satisfying the access policy *T* can decrypt and retrieve the original message M. The Attribute Authority generates the user's private key based on their attribute set S (e.g., age, job title). Private key generation relies on the master key MK and the user's attribute set. For each attribute *i*, a random number $\mathrm{r}_{\mathrm{i}} \in \mathrm{Z}_{\mathrm{p}}$ is chosen, and the user's private key SK is calculated as shown in Formula 4:

$$\mathrm{SK} = \{\mathrm{D_i} = \mathrm{g}^{\alpha}\mathrm{H(i)}^{\mathrm{ri}}, \mathrm{Di}' = \mathrm{g}^{\mathrm{ri}}\}\mathrm{i} \in \mathrm{S} \quad (4)$$

Where $H(i)$ is the hash result for attribute i, ensuring ensuring that each component of the private key is usable only by users with the corresponding attributes. S: The set of attributes of the user, $\mathrm{Di}' = \mathrm{g}^{\mathrm{ri}}$ is another part in the private key, related to attribute i, which ensures the correctness and security of the key. Upon receiving the ciphertext CT, the user decrypts it through their private key SK and the policy tree structure in the ciphertext. The decryption process involves solving each node in the tree to ultimately obtain the original message M. If the user's attribute set S satisfies the access policy T, the shared key s can be reconstructed from the access tree, allowing decryption of the original message M using the following formula: The shared key $\mathrm{e(g,g)}^{\alpha \cdot \mathrm{s}}$ is computed at the root node, allowing the message M to be recovered as shown in Formula 5:

$$\mathrm{M} = \frac{\mathrm{C}}{\mathrm{e(C',D)}^{\mathrm{s}}} \quad (5)$$

Where $\mathrm{e(C',D)}$ denotes the shared part of the key at the root node. This computation completes the decryption process, allowing the user to retrieve the original plaintext message. In conclusion, the CP-ABE algorithm model involves system initialization, encryption, private key generation, and decryption. By setting access policies and utilizing attribute sets to control key distribution, CP-ABE effectively enables data access control in cloud storage, allowing only users with matching attributes to access the data. CP-ABE's attribute-based access control mechanism significantly enhances data security and flexibility in cloud storage environments.

4. Cloud Storage Data Access Control System Design

To ensure user data security and privacy in cloud storage, a third-party audit mechanism is integrated, ensuring reliable data storage and access control through secure message flows and coordinated roles. The system architecture, shown in Figure 3, includes Cloud Users, a Third-Party Auditor (TPA), and Cloud Servers. Each component works together through specific data and message flows to maintain data security [9].

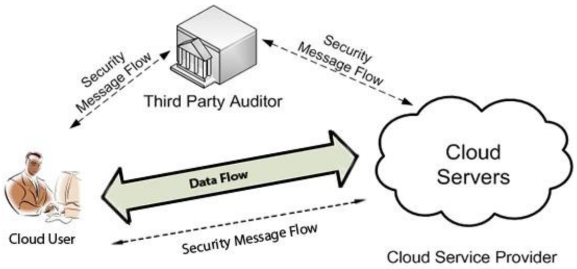

Figure 3. Cloud Storage Data Access Control Architecture Based on Third-Party Auditing.

Cloud Users, which include both Data Owners and Access Requesters, interact with the Cloud Server to either upload or request access to encrypted data. Before storage, data is encrypted, and access control policies are applied to verify identities upon access requests. Encrypted data flows from the Cloud User to the Cloud Server, ensuring protection during transmission. Secure message flows allow Cloud Users to send access policies and authentication information to the Cloud Server, verifying compliance and security. The TPA independently audits data storage and access, ensuring the Cloud Service Provider complies with data owner security requirements. It communicates with the Cloud Server via secure message flows, sending audit requests and receiving logs and data operation records for review. This process enables the TPA to detect unauthorized access or potential security risks, promptly notifying the Cloud User to take preventive actions. Through these independent audits, Cloud Users gain visibility into data storage and access, fostering trust in the cloud service. The Cloud Server manages core tasks related to data storage and access control. Upon receiving secure messages, it verifies the User's access rights based on control policies. If the User's attributes match the policy, access is granted; otherwise, it is denied. The Cloud Server also cooperates with the TPA by providing operational logs for auditing, adhering strictly to security policies, and enhancing data security and transparency. In summary, this system design offers a secure, efficient, and transparent access control solution for cloud storage, enabling safe data management and sharing while maintaining high security standards [10].

5. System Implementation and Performance Analysis

To verify the effectiveness and performance of the cloud storage data access control system based on the CP-ABE algorithm, this experiment tested multiple aspects, including encryption and decryption efficiency, access control verification performance, and the impact of third-party auditing. These tests provided insights into the system's behavior under varying data volumes and access policy complexities, enabling analysis of its stability and scalability. The experimental environment included a 64-bit Ubuntu server (16GB RAM, 2.4GHz quad-core CPU) as the cloud server and several simulated clients for testing user access and auditing functions, where CP-ABE encryption was implemented via the Charm-Crypto library to ensure standardized cryptographic operations. The experiment mainly analyzed encryption and decryption performance, response speed of access control policies, and response time of third-party auditing. The specific data and results are as follows:

Table 1 shows the encryption and decryption times for various data sizes. As the data size increased from 1MB to 200MB, encryption and decryption times grew almost linearly. For 1MB of data, encryption and decryption times were 115ms and 85ms, respectively. When data size reached 100MB, encryption and decryption times were 910ms and 865ms, and for 200MB, both times approached 2 seconds. This trend indicates that the system performs well with small to medium data sizes but encounters performance bottlenecks with larger data sizes, which may impact user experience in high-frequency access scenarios, particularly when multiple users simultaneously request access to large encrypted files.

Table 1. Encryption and Decryption Performance with Different Data Sizes.

Data Size (MB)	Encryption Time (ms)	Decryption Time (ms)
1	115	85
5	135	110
10	250	215
20	340	310
50	460	420
100	910	865
200	1810	1740

Table 2 displays the response times for different access policies of varying complexity. For simple access policies involving a single attribute, the average response time remained below 30 milliseconds. As policy complexity increased, response times rose accordingly. For complex three-attribute and four-attribute policies, response times were around 80ms and 100ms, respectively. Although increased policy complexity imposes additional computational overhead, overall response times remain within acceptable limits, supporting multi-level access control requirements. The system handles "AND" and "OR" combinations with acceptable response times (under 110ms), indicating that it can efficiently support multi-condition access control scenarios.

Table 2. Access Control Verification Performance with Different Policy Complexities.

Policy Complexity	Policy Type	Matching Condition	Response Time (ms)
Simple	Single Attribute	Match	28
Simple	Single Attribute	No Match	22
Moderate	Two Attributes AND	Match	56
Moderate	Two Attributes AND	No Match	50
Complex	Three Attributes AND	Match	81
Complex	Three Attributes OR	Match	76
Complex	Three Attributes OR	No Match	68
Highly Complex	Four Attributes AND	Match	105
Highly Complex	Four Attributes OR	Match	98

Table 3 presents the CPU and memory usage for encryption and decryption across various data sizes. As data size increased, both CPU and memory consumption rose significantly. For small data (e.g., 1MB), CPU usage was relatively low, at 10% for encryption and 8% for decryption. At 100MB, CPU usage for both encryption and decryption reached approximately 70%. Memory usage increased from 20MB to 200MB, indicating substantial resource consumption during large-scale data processing, which could be a limitation in resource-constrained environments.

Table 3. CPU and Memory Consumption for Encryption and Decryption.

Data Size (MB)	Encryption CPU Usage (%)	Decryption CPU Usage (%)	Encryption Memory Usage (MB)	Decryption Memory Usage (MB)
1	10	8	20	18
5	12	10	30	28
10	18	15	45	42
20	25	22	70	65
50	45	40	130	120
100	75	68	200	190

Table 4 records third-party audit response times for various rounds with different audit request numbers. The average response time was around 100ms, with a maximum response time below 120ms. This demonstrates that the TPA maintains good stability under system load, completing data operation reviews promptly with minimal impact on the cloud server. Even with high-frequency audit requests, the system's audit response time does not increase significantly. From the experimental results, the CP-ABE-based cloud storage data access control system performed well in encryption/decryption efficiency, access control verification, and third-party audit response times, making it suitable for most cloud storage application scenarios. Encryption and decryption times are proportional to data size, exhibiting linear scalability, although performance bottlenecks arise with very large data sizes (e.g., above 200MB). Therefore, in scenarios with frequent access to large datasets, algorithm optimization or distributed computing resources may be necessary to enhance performance. In access control verification tests, the system's response time varied with policy complexity. Simple policies were verified quickly, while complex policies required additional time. However, delays of around 100ms remain acceptable for most applications, indicating the system's flexibility and scalability for handling complex access control requirements and supporting fine-grained permission management. The introduction of third-party auditing did not significantly impact system performance. The experiment showed that even during audit request peaks, the TPA maintained an average response time around 100ms, demonstrating good transparency and auditability. The TPA's auditing mechanism provides a security guarantee for data access, allowing users to monitor data usage in real-time and enhancing system trustworthiness. In conclusion, the CP-ABE-based cloud storage data access control system performs excellently in small to medium data environments but may require further optimization for larger data scenarios. The system's access control and third-party auditing mechanisms enhance data security and transparency, allowing users to securely share and manage data in cloud storage environments. Future work could incorporate distributed storage and parallel computing technologies to improve the system's efficiency in handling large datasets, optimize resource consumption, and enhance scalability and adaptability.

Table 4. Third-Party Audit Response Times.

Audit Round	Number of Audit Requests	Average Response Time (ms)	Max Response Time (ms)	Min Response Time (ms)
1	10	102	115	95
2	20	108	120	98
3	30	105	118	97
4	40	107	121	96

6. Conclusion

This research presents a data access control system suitable for cloud storage environments, designed based on the CP-ABE algorithm. By implementing encryption policies, it achieves fine-grained permission management, and the inclusion of a third-party audit mechanism enhances security and transparency. Experimental results demonstrate the system's strong encryption/decryption performance and response speed for access control in small to medium data scales, effectively meeting multi-level access control needs. While certain performance bottlenecks are observed in large data processing, overall, the system proves highly practical for ensuring data security and compliance in cloud storage. Future research could further optimize the algorithm and incorporate distributed computing to enhance efficiency and scalability in large-scale data scenarios.

References

1. R. Cheng et al., "An efficient ECC-based CP-ABE scheme for power IoT," *Processes*, vol. 9, no. 7, p. 1176, 2021, doi: 10.3390/pr9071176.
2. Z. Zhang, W. Zhang, and Z. Qin, "A partially hidden policy CP-ABE scheme against attribute values guessing attacks with online privacy-protective decryption testing in IoT assisted cloud computing," *Future Gener. Comput. Syst.*, vol. 123, pp. 181–195, 2021, doi: 10.1016/j.future.2021.04.022.
3. S. Wang et al., "A fast CP-ABE system for cyber-physical security and privacy in mobile healthcare network," *IEEE Trans. Ind. Appl.*, vol. 56, no. 4, pp. 4467–4477, 2020, doi: 10.1109/TIA.2020.2969868.
4. N. Chen et al., "Efficient CP-ABE scheme with shared decryption in cloud storage," *IEEE Trans. Comput.*, vol. 71, no. 1, pp. 175–184, 2020, doi: 10.1109/TC.2020.3043950.
5. J. Ma et al., "CP-ABE-based secure and verifiable data deletion in cloud," *Secur. Commun. Netw.*, vol. 2021, Art. no. 8855341, 2021, doi: 10.1155/2021/8855341.
6. Z. Zhang and X. Ren, "Data security sharing method based on CP-ABE and blockchain," *J. Intell. Fuzzy Syst.*, vol. 40, no. 2, pp. 2193–2203, 2021, doi: 10.3233/JIFS-189318.
7. M. Taha, H. Ould-Slimane, and C. Talhi, "Smart offloading technique for CP-ABE encryption schemes in constrained devices," *SN Appl. Sci.*, vol. 2, no. 2, p. 274, 2020, doi: 10.1007/s42452-020-2074-z.
8. M. Xie et al., "A CP-ABE scheme based on multi-authority in hybrid clouds for mobile devices," *Future Gener. Comput. Syst.*, vol. 121, pp. 114–122, 2021, doi: 10.1016/j.future.2021.03.021.
9. S. Das and S. Namasudra, "Multiauthority CP-ABE-based access control model for IoT-enabled healthcare infrastructure," *IEEE Trans. Ind. Inform.*, vol. 19, no. 1, pp. 821–829, 2022, doi: 10.1109/TII.2022.3167842.
10. D. V. K. Vengala, D. Kavitha, and A. P. S. Kumar, "Secure data transmission on a distributed cloud server with the help of HMCA and data encryption using optimized CP-ABE-ECC," *Clust. Comput.*, vol. 23, no. 3, pp. 1683–1696, 2020, doi: 10.1007/s10586-020-03114-1.

Disclaimer/Publisher's Note: The views, opinions, and data expressed in all publications are solely those of the individual author(s) and contributor(s) and do not necessarily reflect the views of PAP and/or the editor(s). PAP and/or the editor(s) disclaim any responsibility for any injury to individuals or damage to property arising from the ideas, methods, instructions, or products mentioned in the content.