IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 6, December 2026, pp. 5995~6004
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i6.pp5995-6004

Blockchain-integrated ciphertext-policy attribute-based encryption for secure access control in cloud

Ajay Narasaiah[1], Mohan Hosaagrahara Savalegowda[2], Sumit Gupta[3], Shrihari Melur Ramakrishnappa[1]
[1]Department of Computer Science and Engineering, SJC Institute of Technology, Visvesvaraya Technological University, Belagavi, India
[2]Department of Computer Science and Engineering (Data Science), RNS Institute of Technology, Visvesvaraya Technological University, Belagavi, India
[3]Department of R&D, DeepCognix AI Labs, Bengaluru, India

Article Info

Article history:

Received Jul 9, 2025
Revised Aug 25, 2026
Accepted Sep 20, 2026

Keywords:

Access control
Blockchain
Ciphertext-policy attribute-based encryption
Cloud security
Homomorphic secret sharing
Privacy preservation
Zero-knowledge proof

ABSTRACT

Cloud platforms are now central to storing and processing sensitive information such as electronic health records, which makes confidentiality and controlled access essential design goals. Ciphertext-policy attribute-based encryption (CP-ABE) is a widely adopted primitive that ties data access to user attributes and thereby enforces fine-grained sharing policies. Conventional constructions, however, embed the access policy in plaintext within the ciphertext and rely on costly pairing operations, exposing user attributes and burdening resource-constrained devices. This paper proposes a lightweight CP-ABE framework that couples hidden access policies with an efficient key-generation pipeline. User attributes and policies are anonymized through cryptographic hashing, secret keys are produced using homomorphic secret sharing (HSS) so that the key authority never observes the underlying attributes, and a short Boneh-Lynn-Shacham (BLS) signature protects data integrity. A consortium blockchain with zero-knowledge proofs (ZKP) and smart contracts records attribute issuance, access, and revocation events for tamper-evident auditing. Experiments spanning attribute counts from one to twenty and image sizes of 64 × 64, 128 × 128, and 256 × 256 pixels show lower key-generation, encryption, and decryption times and reduced ciphertext expansion relative to standard CP-ABE, CP-ABE with homomorphic reduction, and lightweight CP-ABE baselines. The design suits internet of things (IoT) and mobile settings where computation and bandwidth are limited.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Ajay Narasaiah
Department of Computer Science and Engineering, SJC Institute of Technology
Visvesvaraya Technological University
Belagavi, Karnataka, India
Email: n.ajay2@gmail.com

1. INTRODUCTION

Cloud computing has shifted from a convenience to the default platform for modern digital services. Providers offer on-demand access to pools of compute, storage, and hosted applications that scale with little administrative work, at a cost model most organizations find attractive. Gartner estimates that worldwide public-cloud spending will grow from roughly 591.8 billion dollars in 2023 to more than 724 billion dollars in 2024 [1], with infrastructure, platform, and software as a service all contributing to the rise. Hospitals, banks, universities, and government agencies now run routine operations on cloud infrastructure.

Much of the research prompted by this growth targets resource management and scheduling, where tasks must be matched to available resources under deadline constraints: a deep reinforcement learning scheduler [2], a hybrid multi-objective optimization method [3], surveys of related techniques [4], [5], and a hybrid dynamic load-balancing scheme [6] are representative. The security and privacy of the data held in multi-tenant; distributed clouds have drawn far less attention. Clouds have garnered much more attention, particularly distributed clouds, whereas tenants have not. The consequences of having financial data, health information, and other personal information stored in clouds are significant, as a misconfigured policy, malicious insider, or even a single unauthorized access can be very costly. In 2023, International Business Machines Corporation (IBM) estimated the average breach cost at $4.45 million, and noted that breaches that occur within cloud environments take longer to contain [7]. These examples are the exposure of over 100 million capital one records on Amazon Web Services (AWS) [8], the leak of approximately 540 million Facebook records from third-party servers [9], and the unsecured storage bucket at Accenture [10] are examples of the risk. Access control should not, however, be inefficient in any way while leaking nothing during policy evaluation, policy transmission, or access authentication. The most flexible of the classical models is attribute-based access control, which allows access based on attributes of the user, the resource, and the environment. Lu *et al.* [11] extended this line of thought to create a blockchain-based system for medical data sharing that incorporates ciphertext-policy attribute-based encryption (CP-ABE), proxy re-encryption, and the interplanetary file system (IPFS) with fine-grained revocation, traceable key distribution, and enhanced-raft consensus.

This paper develops a privacy-preserving data-sharing scheme for cloud systems on a CP-ABE foundation. Its main contributions, each aimed at a limitation of existing designs, are: i) attribute and policy anonymization: user attributes and access policies pass through a cryptographic hash, so identity information never appears during key generation or ciphertext distribution; ii) homomorphic secret sharing (HSS) for key generation: the key generation center (KGC) derives secret keys from anonymized, shared attribute representations instead of plaintext attributes, preserving anonymity and keeping key-generation cost manageable at scale; iii) Boneh-Lynn-Shacham (BLS) signatures for integrity: it is applied to each ciphertext to ensure that the person who owns it is the rightful owner, and to detect unauthorized modification; and iv) blockchain-based auditing and revocation: smart contracts handle revocation and validation, blockchain entries record issues, access and revoking, and zero-knowledge proofs (ZKP) validate attribute possession without revealing values [12].

Xu *et al.* [13] proposed a hybrid scheme of combining role-based access control (RBAC) with identity-based cryptography for cloud storage, where the re-encryption process is performed when data is written to the cloud to accommodate role and policy changes without re-encryption of all stored data. Saxena and Alam [14] combined RBAC with identity-based and broadcast encryption, which allows the use of one ciphertext by many authorized users without use of a public-key infrastructure. In this work, Azbeg *et al.* [15] have built a decentralized remote patient-monitoring system using the combination of proxy re-encryption, an Ethereum proof-of-authority blockchain, and IPFS off-chain storage. Singamaneni *et al.* [16] adopted heavy encryption to the cloud within a quantum key-policy attribute-based encryption (ABE) architecture based on quantum-resistant primitives for constrained internet of things (IoT) devices. Kaur *et al.* [17] combined the immutability of blockchain with IPFS and CP-ABE of electronic health records (EHRs), using Ethereum smart contracts to manage permissions and audit trails. Merlec and In [18] proposed smart-contract-based context-aware access control (SC-CAAC), a smart-contract scheme for Hyperledger Besu which makes access decisions based on real-time context (identity, location, and device state). Qin *et al.* [19] proposed the BMAC on the hyperledger fabric, distributing attribute management among authorities using the Shamir secret sharing scheme and enabling automatic issuance and decryption using smart contracts. Chinnasamy and Deepalakshmi [20] presented hybrid cryptographic access control for electronic health record (HCAC-EHR) that encrypts the record using a modified version of Rivest-Shamir-Adleman (RSA), keeps keys protected with blowfish encryption, and shares them using image steganography and role-based sharing. Chen *et al.* [21] used attribute-based access control, k-anonymity, and bilinear-pairing searchable encryption on a consortium hyperledger fabric network for cross-institution medical data.

2. METHOD

The proposed privacy-preserving auditable access control scheme for cloud data sharing is based on CP-ABE [22] with hidden access policies, and includes two additional components: HSS [11] to improve the key generation efficiency and a consortium blockchain [12] to track all access and attribute events in an immutable manner. The workflow consists of six steps: system initialization, registration and attribute assignment, key generation with HSS, data encryption using BLS signing, blockchain-based auditing, and data decryption using signature verification, which are explained in detail in the following.

2.1. Preliminary

Bilinear maps are the foundation of attribute-based encryption. Assume G_1 and G_T are two multiplicative cyclic groups of order a large prime p, and g is a generator of G_1. Let $e: G_1 \, x \, G_1 \, -> \, G_T$ be a symmetric bilinear map; it is required to satisfy three conditions:

i) Bilinearity: $e(u^m, v^n) = e(u, v)^{mn}$ for every u, v in G_1 and all m, n in Z_p.

ii) Non-degeneracy: $e(g, g)$ is not the identity element of G_T.

iii) Computability: $e(u, v)$ can be computed efficiently for any input pair.

In combination, these features: fine-grained encryption, anonymity and efficient key management, and an auditable trace: secure the major vulnerabilities in cloud sharing. Figure 1 shows the overall architecture.

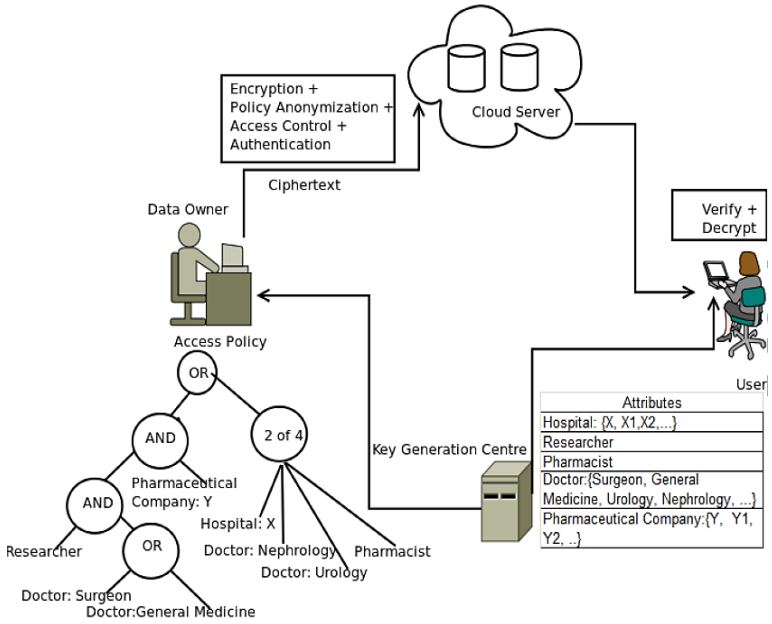

Figure 1. Proposed system architecture for access control in cloud

2.2. Overview of architecture

The general structure of the framework is that a trusted KGC computes users' public keys from their master key and bilinear pairings. All users sign up with an aware access control (AA), which verifies the identity and assigns an attribute group based on their role; each attribute gets hashed before it is submitted to the KGC. Key derivation is then performed over the hash of the attributes, which is homomorphic, thereby avoiding the necessity of the KGC to reconstruct the attribute values and further reducing the time taken for key derivation and eliminating an attribute-leakage path. On the owner side, an access policy is constructed using AND/OR/threshold gates, then anonymized by hash functions and then put into the ciphertext using CP-ABE. The encrypted and signed data are then sent to a cloud server that does not know the data's content or structure. Each access attempt, along with each attribute or policy event, is recorded on a consortium blockchain. Only those attributes that satisfy the hidden policy are allowed to be requested; when they are requested, then the ciphertext is decrypted and verified, and when they are not, then the request is refused and is recorded.

2.3. Proposed data sharing scheme with hidden access policy

While CP-ABE enables the owner to specify who has access to view a ciphertext, the access policy is typically sent as part of the encrypted text, which contains sensitive attributes. The proposed scheme therefore conceals the policy from the user, the KGC, and the CSP via tokenization and a collision-resistant hash function, e.g., SHA-256, and uses a BLS short signature over bilinear pairings to check the identity of the policy owner and the integrity of the ciphertext, followed by five operational phases: setup, KeyGen, SignKey, EncryptSign, and DecryptVerify.

i) Step 1: system initialization. The KGC, a fully trusted authority, bootstraps attribute-based encryption over bilinear pairings on elliptic curves and fixes the global parameters for key distribution, encryption, and verification. It selects two multiplicative cyclic groups G_1 and G_T of large prime order p with a

bilinear map e: $G_1\ x\ G_1 \rightarrow\ G_T$ satisfying the conditions of subsection 2.1. Let $g \in\ G_1$ be the generator of the group, and let $a, b \in\ Z_p$ be the master secret exponents, with $h = g^b, f = g^{\frac{1}{b}}$, and $e(g,g)^a$ used to embed the message securely in the ciphertext. On this basis, the master secret key and public parameters are computed as (1) and (2).

$$Master\ Key\ (MK) = (b, g^a) \tag{1}$$

$$Public\ key: PK = (g, h, f, e(g,g)^a) \tag{2}$$

ii) Step 2: user registration and attribute assignment: in standard ABE, the user attributes reach the key-issuing authority in plaintext, letting it learn sensitive roles or identifiers. Here, the AA verifies each identity, assigns roles, and hashes the attributes before transmission, so the KGC never sees their true values. Let $A_u = \{a_1, a_2, \dots, a_k\}$ be the attribute set of user u, and let H be a secure hash such as secure hash algorithm 256-bit (SHA-256). The anonymized attributes are then defined as (3). These anonymized attributes drive key generation and policy matching without exposing actual values.

$$\tilde{a}_j = H(a_j),\ \forall_j \in A_u \tag{3}$$

iii) Step 3: optimal key generation via HSS: binding keys directly to plaintext attributes harms both privacy and scalability, so the scheme adopts HSS: computation proceeds over shared values, and the KGC builds user keys without knowing the attributes. Its additive homomorphism keeps attribute-key computation efficient under many concurrent key requests. Let $\rho, \rho_j \in Z_p$ be random values for personalization and per-attribute randomness, and let $j \in \tilde{A}_u$, the anonymized attribute set of users u. The secret-key components are then formed as (4).

$$K = g^{\frac{a+\rho}{b}}, K_j = g^{\rho} H(j)^{\rho j}, K_j' = g^{\rho j} \tag{4}$$

The complete secret key of user u is expressed as (5).

$$SK_u = \left(K, \{K_j, K_j'\}_{j \in \tilde{A}_u}\right) \tag{5}$$

iv) Step 4: data encryption with anonymized policies and BLS signature: the owner anonymizes the policy with hash functions before encrypting under a bilinear-map CP-ABE construction, and a BLS short signature on each ciphertext lets a receiver confirm the data source and detect tampering. Let us consider that the policy is expressed as $A = \{a_1, a_2, \dots, a_n\}$, as (6).

$$\tilde{A} = \{H(a_1), H(a_2), \dots\} \tag{6}$$

The encryption process considers: random $z \in \mathbb{Z}_p$, $C' = M.e(g,g)^{az}$, $C = h^z$. For each attribute y. C_y is (7).

$$C_y = g^{q_y(0)}, C_y' = H\big(att(y)\big)^{q_y(0)} \tag{7}$$

Here $q_y(0)$ is the secret share assigned to leaf node y of the access tree and $att(y)$ its attribute label. Signature: hash message $h\ (M)$, private key $x \in \mathbb{Z}_p$ and signature $\sigma = h(M)^x$, where g^x is published for verification. The ciphertext is then generated as (8).

$$CT = \left(\tilde{A}, C, C', \{C_y, C_y'\}, \sigma\right) \tag{8}$$

v) Step 5: blockchain-enabled auditing and policy traceability: a consortium blockchain underpins auditability, recording attribute assignment, revocation, and access attempts as immutable transactions. Smart contracts enforce revocation and policy-update rules, and ZKP confirm attribute ownership without disclosing the attributes. Let us consider that the cloud model transactions are represented as

$Tx = \{issue, access, revoke, update\}$, its hash is updated as $TxHash = H(Tx)$, and the smart contract C enforces $C(Tx) \Rightarrow \frac{Valid}{Reject}$. Similarly, the ZKP is presented as (9).

$$ZKP(\tilde{a}_j) \Rightarrow Proves\ ownership\ of\ a_j\ without\ revealing\ a_j \quad (9)$$

All these events are logged as (10).

$$BlockchainLedger \leftarrow append(TxHash) \quad (10)$$

vi) Step 6: decryption and signature verification: a user who satisfies the anonymized policy reconstructs the encryption randomness by recursive evaluation of the access tree, recovers the plaintext through the CP-ABE decryption relation, and checks the BLS signature; unauthorized users learn nothing, and tampering is exposed. The decryption function is given as (11).

$$DecryptNode(CT, SK, j) = \frac{e(K_j, C_y)}{e(K'_j, C'_y)} = e(g, g)^{\rho_q y(0)} \quad (11)$$

Combining all valid shares gives the encrypted content, and the key must be (12) and (13).

$$e(K, C) = e\left(g^{\frac{a+\rho}{b}}, h^z\right) = e(g, g)^{(a+\rho)z} \quad (12)$$

$$M = \frac{C'}{e(K,C)/e(g,g)^{\rho z}} = M \quad (13)$$

Signature verification requires the following condition (14).

$$Accept\ if\ e(\sigma, g) = e(h(M), g^x) \quad (14)$$

The computational cost of each phase is straightforward to bound. Setup performs a constant number of exponentiations, KeyGen requires $O(k)$ exponentiations for a user holding k attributes since each anonymized share is touched once, and EncryptSign takes $O(n)$ exponentiations for a policy over n attributes plus one signing operation. DecryptVerify dominates with the need for $O(|S|)$ pairings to determine if there is a match in the set S of matched attributes, plus two additional pairings for the BLS check (check to see if the mask is correct), and logging adds a constant cost per event.

3. RESULTS AND DISCUSSION

3.1. Implementation details

The program was developed and tested on a workstation that is equipped with an Intel CPU of the model Core i7-12700k, 32 GB of DDR4 memory, and 64-bit Windows. Python 3.10 was used, along with the additional packages NumPy, PIL, and Matplotlib. The process consists of four steps: the input is encoded as an integer M in Z_{2^n}; the masked version is computed by a symmetric exclusive OR (XOR) with a pseudo-random stream generated from a fixed seed to keep the process deterministic so that repeated runs yield the reported timings; a CP-ABE layer allows decrypting the masked version only if the attribute set matches the policy P; the plaintext is recovered by an inverse XOR operation with the regenerated pseudo-random stream. CP-ABE, not the XOR mask, provides security: It protects the data-encryption key under the anonymized policy. Measuring overhead for key generation, encryption and decryption time, and storage.

3.2. Outcome of proposed model

This scheme was evaluated in terms of encryption time, decryption time, and ciphertext-size overhead, compared to CP-ABE, CP-ABE with homomorphic encryption and reduction (HER), and lightweight CP-ABE, respectively, on images of 64 × 64, 128 × 128, and 256 × 256 pixels. The key-generation time in Figure 2 is shown as a function of the number of attributes (1 to 20).

The encryption times of the three image sizes are shown in Table 1. Standard CP-ABE is the slowest, with a time of 0.12 s at 64 × 64 and 0.50 s at the larger size of 256 × 256, respectively, for CP-ABE HER and lightweight CP-ABE. The same four schemes are plotted in Figure 3 with respect to the number of attributes; as expected, the encryption time increases with the number of attributes. The decryption times for

the proposed scheme are shown in Figure 4, and it is observed that the proposed scheme is competitive as the number of attributes increases.

In contrast, the proposed scheme increases from 0.205 s at 64 × 64 to just 0.155 s at 256 × 256 and has a small fixed cost at low resolutions and scales better with higher resolutions. Table 2 summarizes the decryption time. Standard CP-ABE comes out as the most expensive one, rising from 0.11 to 0.48 s, while the proposed scheme reaches the lowest value of 0.124 s at 256 × 256 image size. The decryption time of the proposed method is slightly higher than that of the other scheme on small images due to the initialization of the cryptographic functions, but is more efficient at larger sizes. Furthermore, the proposed scheme has a compact overhead ratio of 0.85x, while standard CP-ABE, CP-ABE HER, and lightweight CP-ABE are around 1.8×, 1.5×, and 1.3×, respectively, which boosts the efficiency of storage and communication.

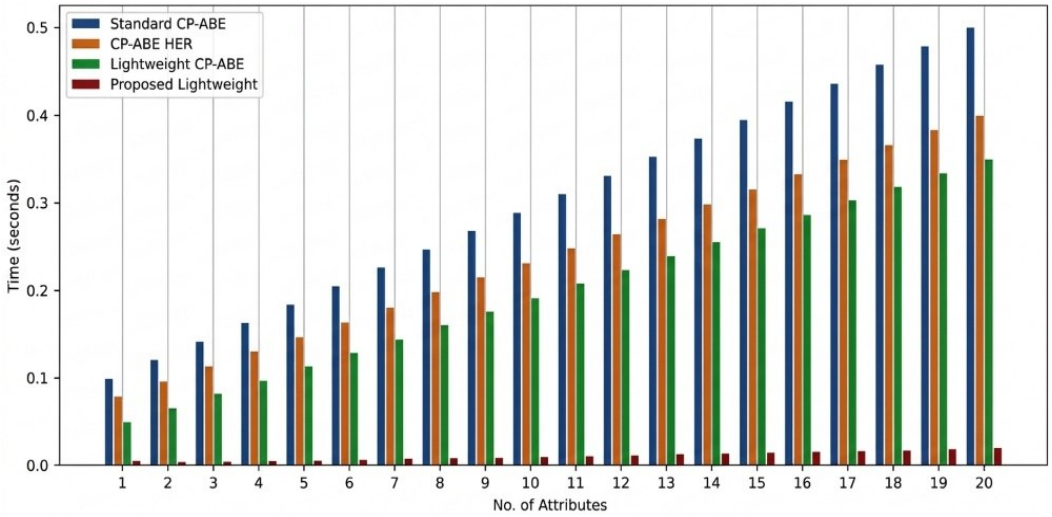

Figure 2. Key generation time vs. number of attributes

Table 1. Encryption time (seconds)

Image size	Standard CP-ABE [22]	CP-ABE HER [23]	Lightweight CP-ABE [24]	Proposed lightweight
64 × 64	0.12	0.10	0.09	0.205
128 × 128	0.25	0.22	0.20	0.255
256 × 256	0.50	0.40	0.35	0.155

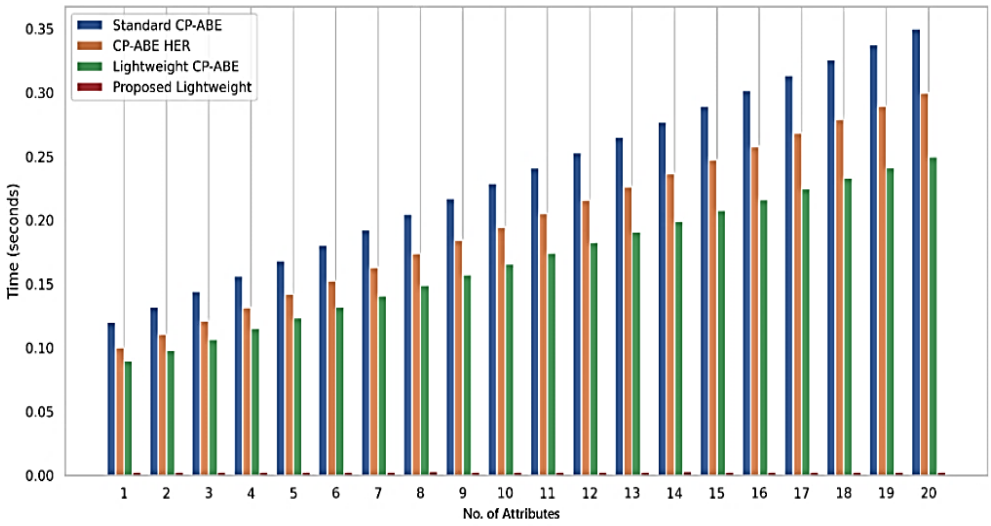

Figure 3. Encryption time vs. number of attributes

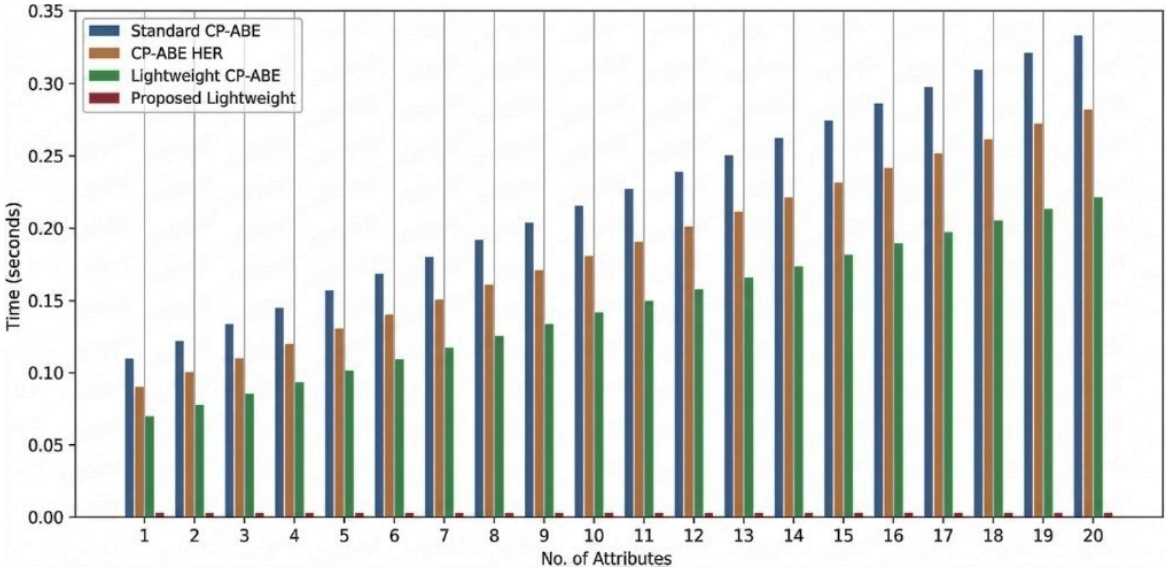

Figure 4. Decryption time vs. number of attributes

Table 2. Decryption time

Image size	Standard CP-ABE [22]	CP-ABE HER [23]	Lightweight CP-ABE [24]	Proposed lightweight
64 × 64	0.11	0.09	0.07	0.164
128 × 128	0.23	0.20	0.18	0.204
256 × 256	0.48	0.38	0.33	0.124

At cloud scale, the overhead of recording issuance, revocation, and access events on the blockchain is 3-5% for key generation and encryption, which is negligible. Smart contracts ensure that revocation is enforced, while ZKP can verify compliance at low cost, without compromising the light-weight design. The evaluation is based on images, but the framework is data-agnostic: the same anonymization, access-control, and auditing mechanisms are applicable to any digital data, since CP-ABE encrypts a symmetric key, not the payload. Similarly, the growth in all three metrics is close to linear with the number of attributes tested from 1 to 20, and requests from different clients are independent, so similar behavior is expected with larger attribute universes and numbers of clients. It's also straightforward to deploy: ciphertexts and signatures are stored in standard cloud object storage, and only the compact hashes are stored in the ledger, keeping the number of chains small.

3.3. Security analysis

The security of the scheme is analyzed about the following attacks: chosen-ciphertext, chosen-plaintext, collusion, replay, and policy hiding attacks [25]. The analysis assumes that the KGC is fully trusted, the AA is honest, and the cloud server is honest but curious; the consortium members are only trusted collectively by consensus. These assumptions are realistic since even if the KGC is compromised, it sees only the hash of the attribute value and can divide the authority role between multiple parties such that no single party has the entire key material.

3.3.1. Data confidentiality

Let $M \in \mathbb{Z}_p$ be the plaintext and PK the public key. The encryption process produces the ciphertext as $CT = Enc(PK, A, M)$ where A is the access structure. The proposed model ensures that an adversary who does not possess attributes satisfying the access policy A cannot distinguish between encryptions of any two plaintexts $M_0, M_1 \in \mathbb{Z}_p$. The confidentiality requirement is (15).

$$\left|\Pr[\mathcal{A}(CT^*) = b] - \frac{1}{2} \leq \epsilon(\lambda)\right| \quad (15)$$

Here, $\mathcal{A}$ represents the polynomial-time adversary, λ is the security parameter, and $\epsilon(\lambda)$ is the negligible function. This assumption is based on the hardness of the decisional bilinear Diffie-Hellman (DBDH) problem.

3.3.2. Collusion resistance

Let $\{SK_1, SK_2, .., SK_t\}$ be the private keys corresponding to attribute sets $\{S_1, S_2, \dots, S_t\}$, none of which alone satisfies the policy A. Even if their holders collude, they cannot decrypt unless the combined attributes satisfy the policy, as shown (16). Randomizing ciphertext components and binding key parts to individual attributes makes key aggregation infeasible, so collusion cannot yield unauthorized decryption.

$$S' = \bigcup_{i=1}^{t} S_i \nVdash A \Rightarrow Decrypt(CT, \{SK_i\}) \rightarrow \perp \quad (16)$$

3.3.3. Policy hiding

Traditional CP-ABE exposes the access policy within the ciphertext. In contrast, our approach anonymizes the policy using cryptographic obfuscation. Let $A(x_1, x_2, \dots, x_n)$ represent a monotonic Boolean function for access control. The obfuscated form is (17). Where π is a random permutation function, and $H(A)$ is the hashed policy structure. This prevents an adversary from learning the original policy, thereby ensuring access pattern privacy.

$$A' = f(A, \pi), CT = (C', \pi(H(A))) \quad (17)$$

3.3.4. Chosen-ciphertext attack resistance

The encryption process involves a hybrid mechanism as defined in (18). Where k is the symmetric key, $CT_{ABE}(k) = Enc_{ABE}(PK, A, k)$ is the CP-ABE of the encryption key, $Enc_z(M) = k \oplus PRG(seed)$ represents the symmetric encryption using a pseudo-random generator. As the key k is encrypted under a CPA-secure ABE scheme and the adversary cannot access a decryption oracle post-challenge, the model ensures security against adaptive chosen-ciphertext attacks under the random oracle model.

$$CT = \left(CT_{ABE}(k), Enc_z(M)\right) \quad (18)$$

3.3.5. Replay attack mitigation

To prevent replay attacks, each encryption instance includes a unique nonce z as (19). Encrypting the same plaintext repeatedly therefore yields distinct ciphertexts under different seeds, which defeats replay attempts.

$$CT = \left(CT_{ABE}(k), z, Enc_z(M)\right), Enc_z(M) = M \oplus PRG(z) \quad (19)$$

4. CONCLUSION

The study proposed a lightweight CP-ABE framework to provide access control security and performance while protecting confidentiality of cloud data. It anonymizes attributes and policies, generates keys based on HSS, ensures integrity with BLS signatures, and audits access on a consortium blockchain. It performed better than standard, HER, and lightweight CP-ABE in terms of overhead and cryptographic operation speed over different attribute sets and data sizes, especially in the encryption time and ciphertext expansion. Going forward, communication overhead will be reduced, dynamic policy updates and revocation will be explored, and multi-tenant, latency-aware deployment on production clouds will be done. Access policies that are optimized through artificial intelligence (AI) and have components that detect anomalous requests are another extension, with the cryptographic layer retaining the final veto.

FUNDING INFORMATION

Authors state no funding was involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Ajay Narasaiah	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
Mohan Hosaagrahara Savalegowda	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓		
Sumit Gupta	✓	✓	✓	✓				✓	✓	✓	✓	✓		
Shrihari Melur Ramakrishnappa	✓	✓	✓	✓				✓	✓	✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The author declares that there are no known conflicts of interest associated with this publication. There are no financial or personal relationships that could inappropriately influence or bias the content of this work.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] N. Ajay, H. S. Mohan, I. S. Rajesh, and S. Gupta, "A deep learning and blockchain-based security framework for cloud data protection," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 29, no. 6, pp. 2553–2587, 2026, doi: 10.47974/JDMSC-2823.

[2] K. Siddesha, G. V. Jayaramaiah, and C. Singh, "A novel deep reinforcement learning scheme for task scheduling in cloud computing," *Cluster Computing*, vol. 25, no. 6, pp. 4171–4188, Dec. 2022, doi: 10.1007/s10586-022-03630-2.

[3] A. N. Malti, M. Hakem, and B. Benmammar, "A new hybrid multi-objective optimization algorithm for task scheduling in cloud systems," *Cluster Computing*, vol. 27, no. 3, pp. 2525–2548, Jun. 2024, doi: 10.1007/s10586-023-04099-3.

[4] G. Zhou, W. Tian, R. Buyya, R. Xue, and L. Song, "Deep reinforcement learning-based methods for resource scheduling in cloud computing: a review and future directions," *Artificial Intelligence Review*, vol. 57, no. 5, Apr. 2024, doi: 10.1007/s10462-024-10756-9.

[5] N. Devi *et al.*, "A systematic literature review for load balancing and task scheduling techniques in cloud computing," *Artificial Intelligence Review*, vol. 57, no. 10, Sep. 2024, doi: 10.1007/s10462-024-10925-w.

[6] R. Anitha, "A novel virtual machine migration model for optimizing cloud resource utilization," *International Journal of Computational Intelligence in Engineering*, vol. 1, no. 3, pp. 22–30, 2026.

[7] M. Elgan, "Cost of a data breach: the healthcare industry," IBM.com, Aug. 2024. Accessed: Aug. 2, 2026. [Online]. Available: https://www.ibm.com/think/insights/cost-of-a-data-breach-healthcareindustry

[8] R. Harpur, "AWS data breach: lessons from 4 high-profile breaches," blackfog.com, 2025. Accessed: Aug. 2, 2026. [Online]. Available: https://www.blackfog.com/aws-data-breach

[9] BBC, "Data on 540 million Facebook users exposed," BBC.com, Apr. 2019. Accessed: Aug. 2, 2026. [Online]. Available: https://www.bbc.com/news/technology-47812470

[10] T. Maurer and G. Hinck, "Cloud security: a primer for policymakers", Working Paper, Carnegie Endowment For International Peace, 2020. [Online]. Available: https://assets.carnegieendowment.org/static/files/Maurer_Hinck_Cloud_Security-V3.pdf

[11] Y. Lu, T. Feng, C. Liu, and W. Zhang, "A blockchain and CP-ABE based access control scheme with fine-grained revocation of attributes in cloud health," *Computers, Materials & Continua*, vol. 78, no. 2, pp. 2787–2811, 2024, doi: 10.32604/cmc.2023.046106.

[12] Z. Ren, E. Yan, T. Chen, and Y. Yu, "Blockchain-based CP-ABE data sharing and privacy-preserving scheme using distributed KMS and zero-knowledge proof," *Journal of King Saud University-Computer and Information Sciences*, vol. 36, no. 3, Mar. 2024, doi: 10.1016/j.jksuci.2024.101969.

[13] J. Xu, Y. Yu, Q. Meng, Q. Wu, and F. Zhou, "Role-based access control model for cloud storage using identity-based cryptosystem," *Mobile Networks and Applications*, vol. 26, no. 4, pp. 1475–1492, 2021, doi: 10.1007/s11036-019-01484-4.

[14] U. R. Saxena and T. Alam, "Role based access control using identity and broadcast based encryption for securing cloud data," *Journal of Computer Virology and Hacking Techniques*, vol. 18, no. 3, pp. 171–182, Sep. 2022, doi: 10.1007/s11416-021-00402-1.

[15] K. Azbeg, O. Ouchetto, and S. J. Andaloussi, "Access control and privacy-preserving blockchain-based system for diseases management," *IEEE Transactions on Computational Social Systems*, vol. 10, no. 4, pp. 1515–1527, Aug. 2023, doi: 10.1109/TCSS.2022.3186945.

[16] K. K. Singamaneni, A. K. Budati, and T. Bikku, "An efficient Q-KPABE framework to enhance cloud-based IoT security and privacy," *Wireless Personal Communications*, Mar. 2024, doi: 10.1007/s11277-024-10908-8.

[17] J. Kaur, R. Rani, and N. Kalra, "Attribute-based access control scheme for secure storage and sharing of EHRs using blockchain and IPFS," *Cluster Computing*, vol. 27, no. 1, pp. 1047–1061, Feb. 2024, doi: 10.1007/s10586-023-04038-2.

[18] M. M. Merlec and H. P. In, "SC-CAAC: a smart-contract-based context-aware access control scheme for blockchain-enabled IoT systems," *IEEE Internet of Things Journal*, vol. 11, no. 11, pp. 19866–19881, 2024, doi: 10.1109/JIOT.2024.3371504.

[19] X. Qin, Y. Huang, Z. Yang, and X. Li, "A blockchain-based access control scheme with multiple attribute authorities for secure cloud data sharing," *Journal of Systems Architecture*, vol. 112, 2021, doi: 10.1016/j.sysarc.2020.101854.
[20] P. Chinnasamy and P. Deepalakshmi, "HCAC-EHR: hybrid cryptographic access control for secure EHR retrieval in healthcare cloud," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 2, pp. 1001–1019, Feb. 2022, doi: 10.1007/s12652-021-02942-2.
[21] Y. Chen, L. Meng, H. Zhou, and G. Xue, "A blockchain-based medical data sharing mechanism with attribute-based access control and privacy protection," *Wireless Communications and Mobile Computing*, vol. 2021, no. 1, Jan. 2021, doi: 10.1155/2021/6685762.
[22] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE Symposium on Security and Privacy (SP '07)*, May 2007, pp. 321–334, doi: 10.1109/SP.2007.11.
[23] X. Yang, W. Li, and K. Fan, "A revocable attribute-based encryption EHR sharing scheme with multiple authorities in blockchain," *Peer-to-Peer Networking and Applications*, vol. 16, no. 1, pp. 107–125, 2023, doi: 10.1007/s12083-022-01387-4.
[24] X. Yao, Z. Chen, and Y. Tian, "A lightweight attribute-based encryption scheme for the internet of things," *Future Generation Computer Systems*, vol. 49, pp. 104–112, Aug. 2015, doi: 10.1016/j.future.2014.10.010.
[25] Y. Bai, K. Fan, K. Zhang, H. Li, and Y. Yang, "CR-FH-CPABE: secure file hierarchy attribute-based encryption scheme supporting user collusion resistance in cloud computing," *IEEE Internet of Things Journal*, vol. 11, no. 10, pp. 17727–17739, May 2024, doi: 10.1109/JIOT.2024.3358745.

BIOGRAPHIES OF AUTHORS



Ajay Narasaiah is an assistant professor of Computer Science and Engineering at SJC Institute of Technology, Chickballapur, with more than fifteen years of teaching experience. He holds an M.Tech. from Visvesvaraya Technological University (VTU), Belagavi, and has completed his Ph.D. from VTU. His research interests include cloud computing, AI, and big data. He has reviewed research papers for IEEE conferences and has served as a session chair in two conferences. He has published two patents, one design patent, and one book. He can be contacted at email: n.ajay2@gmail.com.



Mohan Hosaagrahara Savalegowda holds a Ph.D. in Computer Science and Engineering from Dr. MGR University, Chennai, and serves as professor and head of the Departments of CSE (Data Science) and AI and DS at RNS Institute of Technology, Bengaluru. He has served in several reputed institutions including KIT Tiptur, SIT Tumkur, SJBIT, and NHCE Bangalore. He is the recipient of several accolades, including the Best Teacher Award (SJBIT, 2008), Dr. Abdul Kalam Lifetime Achievement Award (2017), and excellence in Research Award by NESIN (2021). His interests include cryptography and network security, image processing, and cloud computing. He can be contacted at email: mohan_kit@yahoo.com.



Sumit Gupta is a researcher in machine learning, deep learning, and AI, with expertise spanning cloud computing, networking, image processing, optimization strategies, and data mining. He has contributed significantly to the development of intelligent systems and scalable algorithms. He heads the research in Department at Xsys Software Technologies, Bengaluru, and is director of DeepCognix AI Labs, a deep-tech startup based in Bengaluru, India. He can be contacted at email: sumit2k4@gmail.com or sumit@deepcognix.com.



Shrihari Melur Ramakrishnappa is an associate professor of Computer Science and Engineering at SJC Institute of Technology with over fifteen years of academic experience, holding a Ph.D., M.Tech., and B.E. from VTU. He has published more than twenty-two Scopus-indexed papers and authored a book on computer networks and cryptography. He has organized and participated in several FDPs, workshops, and conferences, and actively contributes to institutional roles including IQAC, NAAC, NBA, BOE, and Start-Up coordination. His research interests include cloud security, blockchain, IoT, and machine learning. He can be contacted at email: shrihari.mr@gmail.com.