Blockchain-based CP-ABE System Using ECC for Cloud Storage

Walde Rajesh Baliram*, Bashir Alam, M. N. Doja

Department of Computer Engineering, Jamia Millia Islamia, New Delhi

* Corresponding author

doi: https://doi.org/10.21467/proceedings.7.6.59

ABSTRACT

Cloud storage has revolutionized data storage and access, offering a convenient and scalable solution for storing large amounts of data. However, it also raises issues related to security and privacy, especially after storing the sensitive information on remote servers. In this context, attribute-based encryption (ABE) has added substantial consideration by means of a solution to enforce precise access control on encrypted information. A notable form of ABE is Ciphertext-Policy Attribute-Based Encryption (CP-ABE), allowing the data owner to state access rules depends on attributes (such as roles, permissions, or other user-specific factors). The application of Elliptic Curve Cryptography (ECC) in CP-ABE enhances the system's efficiency and security. On top of this, integrating blockchain technology can provide decentralized and immutable record-keeping, which enhances the transparency, trust, and integrity of the encryption and decryption processes. A Blockchain-based CP-ABE system using ECC provides a robust, secure, and decentralized solution for managing access control to encrypted cloud storage. The integration of blockchain ensures transparency, immutability, and auditability, while ECC offers significant security with lower computational burden. By combining these technologies, we can build an effective and safe framework for protecting sensitive information in cloud environments, ensuring that access to and decryption of the information is restricted to authorized users only based on well-defined access policies.

Keywords: Blockchain, CP-ABE, ECC, LSSS.

1 Introduction

Data is highly valuable and can be presented in various formats. Storing large quantities of data on local devices may lead to maintenance issues. To alleviate storage demands on their devices, many people choose to utilize cloud storage. The widespread practice of saving information online enables individuals to effortlessly access and share data via the internet from any location. Cloud storage has transformed the way we store and access data by providing an easy and scalable option for storing substantial amounts of data [1]. Despite these advantages, concerns regarding data security in cloud storage persist. When users upload data into the cloud, they practically lose control over it, potentially exposing it to various security threats. Presently, cloud storage platforms suffer from weaknesses like centralized storage, which may put data security at risk. Moreover, user privacy may be compromised by relying on trusted third parties. Safeguarding data and preventing cloud service providers (CSP) from accessing it is essential, as these providers may not be entirely reliable, and data control could be out of the owner's hands. A common approach in distributed computing environments is encrypting data prior to uploading it to cloud storage using symmetric and asymmetric encryption techniques. In order to use symmetric encryption, the data owner and data user can create a shared session key in advance. However, identifying potential data consumers can be challenging when sharing data across systems. Furthermore, the asymmetric key encryption method requires obtaining public keys for each data user, performing multiple data encryptions, and storing various encrypted versions of the information.

In this context, the attribute-based encryption (ABE) system implements fine-grained access control for encrypted data, In ABE, the access policy is stated by the data owner through a Boolean formula depending on attribute sets [2]. The authority issues unique private keys to users corresponding to their specific attributes. Decryption is possible when the private key's attributes precisely match the ciphertext's Boolean formula. ABE system has two variants: Key-policy attribute-based encryption (KP-ABE) and Cyphertext-

Proceedings DOI: 10.21467/proceedings.7.6; Series: AIJR Proceedings; ISSN: 2582-3922; ISBN: 978-81-989164-9-5

policy attribute-based encryption (CP-ABE). In the KP-ABE system, "the attributes set is connected to ciphertext, whereas access policy is linked to the secret key and the decryption takes place only if the attributes of the ciphertext match the access policy of the secret key" [3]. Conversely, "in the CP-ABE system, the access policy is linked with ciphertext, whereas the attribute set connected with the secret key and the ciphertext is decrypted only if the secret key satisfies the ciphertext's access policy" [4]. ABE employs one-to-many encryption, allowing ciphertext encryption for multiple users, which is particularly suitable for managing access to cloud-stored data [5].

Blockchains, which are decentralized digital ledgers, are designed to resist tampering and provide evidence of any unauthorized alterations. Modern cryptocurrencies have emerged from combining various computing concepts and blockchain technology utilizing cryptographic methods for protection rather than relying on centralized systems or entities. Bitcoin, the first cryptocurrency, introduced blockchain technology [6]. Blockchain security encompasses the efforts and technologies implemented to safeguard blockchain networks. These protective measures are essential in preventing unauthorized access and fraudulent activities. The robust security protocols employed in blockchain technology enable users to trust in the security of digital transactions and data integrity. This security framework enhances confidence and encourages the adoption of blockchain applications across various industries.

2 Related Work

Cloud computing provides various services to users through a usage-based pricing model while keeping expenses low. However, the storage of data in cloud systems raises significant privacy and security issues. Numerous studies have been published that delve into the security challenges associated with cloud storage, methods for encrypting data, and blockchain technology applications. An in-depth examination of security concerns related to various aspects impacting cloud computing was conducted by [7]. [8] performed a comprehensive review on data security and privacy issues, encryption tools, and potential protective measures in cloud storage platforms. [9] investigated advancements in research concerning privacy security issues in cloud computing, focusing on different privacy security protection technologies. How the security of cloud-based data storage can be enhanced by blockchain technology explored by [10]. [11] studied the research trajectories of the blockchain's application in cloud computing. [12] proposed a security structure for blockchain-enabled decentralized cloud storage, enabling users to encrypt and divide their files into segments, which are then randomly distributed across a peer-to-peer network with available storage capacity. In a similar vein, [13] established a hybrid algorithm built on blockchain that involves encrypting data using AES and ECC prior to cloud storage upload, a unique digital signature is generated on the client side, which can be stored on a decentralized block set.

Various scholars have utilized ABE schemes incorporating ECC and bilinear pairing techniques to encrypt data prior to cloud storage. To expand the security and effectiveness of CP-ABE for cloud-based data-sharing systems, [14, 15] suggested a cooperative key management protocol. [15] suggested an AND gate-based CP-ABE scheme, featuring constant-size secret keys and offering cost-effective encryption and decryption solutions using Elliptic Curve Cryptography (ECC). [16] created an attribute-enabled data-sharing system for cloud computing, addressing the key escrow problem in CP-ABE systems. [17] introduced Enhanced CP-ABE, a method designed to protect, distribute, and control access to cloud-stored data. [18] suggested a data access control system that is pairing-free and built on CP-ABE and ECC, employing simple scalar multiplication rather than complicated bilinear pairing that cuts the computational burden. [19] presented an ECC-based CP-ABE system that uses scalar multiplication in place of bilinear pairing and utilizes edge devices for decryption.

Various studies have utilized blockchain technology in conjunction with Attribute-Based Encryption (ABE) schemes founded on Elliptic Curve Cryptography (ECC). [20] established a blockchain system built on java for cloud storage, incorporating CP-ABE through bilinear mapping cryptography to ensure a secure system. In another study for Electronic Health Record (EHR), [21] projected a blockchain-based multi-authority ABE system sharing in edge computing and featuring outsourced decryption. Similarly, [22] introduced a

Proceedings DOI: 10.21467/proceedings.7.6
ISBN: 978-81-989164-9-5

Series: AIJR Proceedings
ISSN: 2582-3922

blockchain-enabled multi-authority ABE for EHR sharing, where outsourced decryption and hidden access policies could be verified, with the blockchain storing validation parameters and enabling data users to confirm third-party decryption accuracy. [23] presented an efficient and secure data-sharing scheme that merged ABE, blockchain, and IPFS. Additionally, [24] proposed a decentralized ABE system for cloud environments with blockchain technology, facilitating detailed searches and protecting sensitive information.

The studies examined predominantly employed the attribute authority to create and distribute a secret key to data users after attribute verification. However, since the attribute authority is not entirely trustworthy within the system, the process of verifying data user attributes could be compromised. To address this, we suggest implementing a blockchain-enabled CP-ABE system, wherein blockchain can store data user attribute sets along with the secret keys for each attribute. In this proposed framework, the blockchain issues the secret key to the data user only after thoroughly verifying their attributes. The security of the secret key is ensured through blockchain technology. The paper's structure is as follows: Section 2 discusses the related work, Section 3 covers the preliminaries, Section 4 outlines the proposed system, and Section 5 provides the security analysis. The paper concludes with Section 6.

3 Preliminaries

3.1 Blockchain

"A blockchain consists of a series of interconnected blocks that serve as a comprehensive record of transactions, functioning similarly to a traditional public ledger. Each block's header contains the hash of the preceding block, creating a parent-child relationship between blocks. The first block in a blockchain, known as the genesis block, has no preceding blocks" [25].

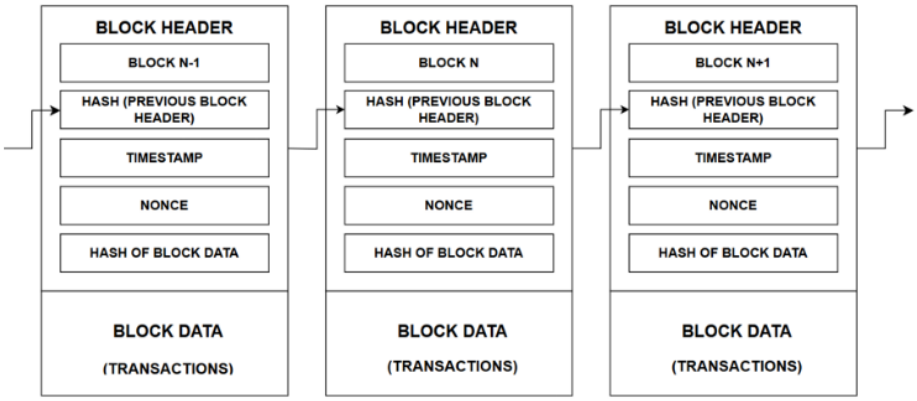

Fig 1. Chain of Blocks

Block: In blockchain technology, a block is comprised of two main elements: a header and data. The metadata about the block is contained in the header section. The data section includes a verified and legitimate set of transactions that have been presented to the blockchain network.

Chain of Blocks: The blockchain is formed by linking the blocks where each block contains a hash digest of the preceding block. Any modification to a previously published block would alter its hash, causing a ripple effect of changed hashes in subsequent blocks due to their incorporation of the previous block's hash. This mechanism allows for easy identification and rejection of tampered blocks. A chain of blocks [26] is illustrated in Fig. 1.

3.2 Notation/Abbreviation and Description

The notations and abbreviations used in this paper are listed in Table 1.

Proceedings DOI: 10.21467/proceedings.7.6
ISBN: 978-81-989164-9-5

Series: AIJR Proceedings
ISSN: 2582-3922

Table 1: Notation/Abbreviation and Description

Notation/Abbreviation	Description
GA	Global Authority
AA	Attribute Authority
CSP	Cloud Service Provider
DO	Data Owner
DU	Data User
BC	Blockchain
LSSS	Linear Secret Sharing Scheme
ECC	Elliptic Curve Cryptography
ECDLP	Elliptic Curve Discrete Logarithm Problem
RSA	Rivest Shamir Adleman
PP	Public parameters
MSK	Master Secret Key
PK	Public Key
SK	Secret key / Private key
M	Message
CT	Ciphertext
$\mathbb{A}$	Access structure
$\prod$	Linear Secret Sharing Scheme
$\mathbb{Z}p$	Set of integers of prime order p
$\mathbb{Z}r$	Set of integers of prime order r
E	Elliptic curve
GF(q)	Finite field of order q
G	Generator point
Q	Point on the elliptic curve
λ	Security parameter
S	Collection of attributes
H	Hash function

3.3 CP-ABE Algorithm

Data confidentiality and access control are frequently protected using encryption technology. However, securing data access control remains a complex task. In 2007, Bethencourt introduced a novel encryption technique known as ciphertext-policy attribute-based encryption (CP-ABE) that combines data encryption and access control to enhance data protection [4]. CP-ABE link the user's secret key to specific attributes and associates the ciphertext with an access structure. To successfully decrypt the ciphertext, users are required to have an attribute set that aligns with the access policy of the data owner. Based on the individual attributes, the user can acquire the necessary secret key from the attribute authority. CP-ABE can assist in delivering precise access control in untrustworthy cloud storage environments, as users can only access data files if their attributes align with the access policies linked to the files [27].

3.4 Access Structure

"Let $\{P_1, P_2, ..., P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P1, P2, ..., Pn\}}$ is monotone if $\forall$ B, C: if B $\in$ A and B $\subseteq$ C, then C $\in$ A. An access structure is a collection $\mathbb{A}$ of non-empty subsets of $\{P_1, P_2, ..., P_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{P_1, P_2, ..., P_n\}} \setminus \{\}$. The sets contained in $\mathbb{A}$ are referred as authorized sets, whereas those that are not contained in $\mathbb{A}$ are referred as unauthorized sets" [28].

Proceedings DOI: 10.21467/proceedings.7.6
ISBN: 978-81-989164-9-5

Series: AIJR Proceedings
ISSN: 2582-3922

For instance, the boolean expression $A \wedge ((B \vee C) \wedge (D \vee E))$ signifies that an individual decrypting the encoded message must possess specific attributes: either A, B, D or A, B, E or A, C, D or A, C, E. This logical structure could be visually characterized as an access tree that offers a more intuitive understanding, as illustrated in Fig 2.

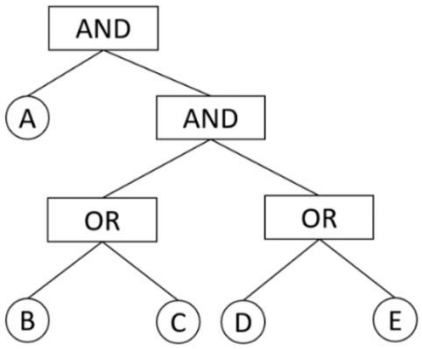

Fig 2. Access tree for boolean formula A ∧ ((B ∨ C) ∧ (D ∨ E))

Additionally, [28] state that by applying standard methods the monotonic access structure could be transformed into LSSS representation.

3.5 Linear Secret-Sharing Scheme (LSSS)

"A linear secret sharing scheme ($\prod$) for a set of parties P (over $\mathbb{Z}p$) must satisfy the following conditions:

1) Each party's share generates a vector over $\mathbb{Z}p$.
2) There is a matrix A referred as the share-generating matrix for $\prod$. The matrix A consists of ℓ rows and n columns. For all x = 1, …., ℓ, the xth row of A is specified by the party $\rho(x)$ (ρ is a function from {1, …, ℓ} to P). Consider the column vector v = (s, r_2,…, r_n), where s $\in \mathbb{Z}p$ is the secret to share and r_2, …, $r_n \in \mathbb{Z}p$ are selected randomly, then A.v is the vector of ℓ shares of the secrets according to $\prod$. The share (A.v)x belongs to the party $\rho(x)$." [28]

Likewise, the linear reconstruction property is fulfilled by the linear secret-sharing scheme. "Consider $\prod$ is an LSSS for access structure $\mathbb{A}$. Let S represent an authorized set and I $\subseteq$ {1, …, ℓ} where I = {x| $\rho(x) \in$ S}. Then, the vector (1,0,…,0) is in the span of rows of A indexed by I, and there exist constants {$\omega \in \mathbb{Z}p$}x $\in$ I such that, for any valid shares {λ}x of a secret s according to $\prod$, we have $\sum$ x $\in$ I $\omega x \lambda x = s$. The constants {ωx} can be determined in polynomial time based on the size of the share-generation matrix" [28].

"The Lewko and Waters algorithm creates a matrix using an access tree that represents a monotonic logic formula as input.

- The access tree's leaf nodes correspond to attributes, while its non-leaf nodes represent AND or OR gates.
- The resulting matrix has several rows equal to the leaf nodes in the access tree.
- As the algorithm moves through the access tree, it assigns labels to the nodes.
- If the parent node is an OR gate labelled with vector v, the algorithm labels both child nodes as v and keeps counter c unchanged.
- If the parent node is an AND gate labelled with a vector v, the algorithm pads v with 0 at the end to change its length to c, and then the algorithm labels its right child node with the vector v || 1. || denotes concatenation here and the left child node with the vector (0, …, 0) || -1, where the length of (0, …, 0) is c, increases the value of c by 1." [29]

Proceedings DOI: 10.21467/proceedings.7.6
ISBN: 978-81-989164-9-5

Series: AIJR Proceedings
ISSN: 2582-3922

Fig 3 demonstrates the algorithm's method for labelling a tree. The rows of a matrix known as LSSS are formed by the vectors found in the tree's leaf nodes. The algorithm extends shorter vectors with zeros to match the length of the longest vector if the vectors are not of equal length. By employing this algorithm, an LSSS matrix can be generated from the access tree shown in Figure 3.

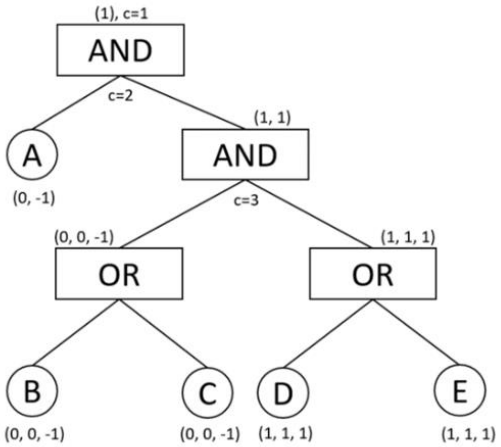

Fig 3. Label the access tree to generate an LSSS matrix.

The LSSS matrix that was produced followed the procedure outlined earlier.

$$M = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & -1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{matrix} \rho(1) = A \\ \rho(2) = B \\ \rho(3) = C \\ \rho(4) = D \\ \rho(5) = E \end{matrix}$$

ρ assigns attributes $A, B, C, D, and\ E$ to each row of the matrix. When a set S of attributes fulfils the LSSS, it indicates that the vector $(1, 0, \dots, 0)$ is within the rows span of the matrix that corresponds to the attributes in S.

For example, consider a file that is encrypted for "PhD students in the Electronics department" and "PG students in the Electronics department". This access policy can be depicted using an LSSS matrix where the attributes "PhD student", "PG students", and "Electronics Department" are represented by rows of the matrix. A user is able to decrypt the file if they hold the attributes "PhD student" and "Electronics Department" or "PG Student" and "Electronics Department". A user who is an UG student in the Electronics department is unable to decrypt the file as they lack the required attributes [30].

3.6 Elliptic Curve Cryptography (ECC)

In contrast to RSA, the ECC provides similar security with reduced key sizes. In 1985, the idea of ECC was presented by Neal Koblitz and Victor Miller [19]. The mathematical representation of an elliptic curve is given by:

$y^2 = x^3 + ax + b$, where $4a^3 + 27b^2 \neq 0$

The security of ECC is founded on the ECDLP that involves determining the logarithm of a point on an elliptic curve within a finite field. Determining the discrete logarithm of a random element on an elliptic curve relative to a specific base point is computationally difficult.

Let E be an elliptic curve defined over a finite field GF(q) of order q. Let G be a generator of the group with order r. For the given point Q on an elliptic curve such that $Q = kG \ \ where\ k \in \mathbb{Z}r$, it is almost

impossible to find an integer k in polynomial time. The ECC encryption protocol comprises three main steps:

- Step 1: The plaintext is mapped to point Q on an elliptic curve.
- Step 2: The encryption protocol is implemented by two parties.
- Step 3: To retrieve the plaintext by mapping point Q on the elliptic curve.

For example, Alice and Bob carry out the following procedure [18]:

1) **Key generation:**
   a) Initially, Alice and Bob agreed to utilize an identical elliptic curve, given by $y^2 = x^3 + ax + b(\text{mod } p)$ with a common generator point G.
   b) For her private key, Alice selects an integer $n_a \in \mathbb{Z}_p$ and calculates her public key by multiplying this value with the point G, resulting in $PK_a = n_aG$.
   c) For his private key, Bob selects an integer $n_b \in \mathbb{Z}_p$ and calculates his public key by multiplying this integer with the point G, resulting in $PK_b = n_bG$.
2) **Encryption:** For the encryption of Q, Alice initiates the process by choosing a random integer k $\in \mathbb{Z}_p$. She then calculates two components of the ciphertext: $C_1 = kG$ and $C_2 = Q + kPK_b$. Following these computations, Alice transmits both C1 and C2 to Bob.
3) **Decryption:** For decryption, Bob initially multiply C_1 with his private key nb. He then subtracts this result from C_2. To obtain the plaintext, Bob maps point Q onto the elliptic curve.

$$\begin{aligned} C_2 - n_bC_1 &= (Q + kPK_b) - n_b(kG) \\ &= (Q + kn_bG) - n_bkG \\ &= Q \end{aligned}$$

3.7 System Model

Typically, a CP-ABE system comprises of four algorithms [5]:

1) **Setup $(\boldsymbol{\lambda}) -> (\boldsymbol{PP}, \boldsymbol{MSK})$:** Initially, the setup algorithm is takings $\boldsymbol{\lambda}$ as input to generate PP and MSK as output.
2) **Encryption $(\boldsymbol{PP}, \boldsymbol{M}, \mathbb{A}) -> \boldsymbol{CT}$:** The encryption algorithm utilizes PP and $\mathbb{A}$ to encrypt M to generate CT.
3) **Key Generation $(\boldsymbol{MSK}, \boldsymbol{S}) -> \boldsymbol{SK}$:** The key generation algorithm utilizes the MSK and S to generate SK.
4) **Decryption $(\boldsymbol{PP}, \boldsymbol{CT}, \boldsymbol{SK}) -> \boldsymbol{M}$:** To decrypt the message, the decryption algorithm utilizes PP, CT with A, and SK that corresponds to S. If the S meets A, then the CT is decrypted by the algorithm to get the original message M.

3.8 Security Model

In CP-ABE, an adversary is allowed by the security framework to obtain private keys that cannot successfully decrypt the intended ciphertext. This system links the ciphertexts with access structures and the private keys with attributes. According to the security model [5], "an adversary can choose a challenge for encrypting an access structure A* and request any private key S that does not meet the requirements of S*".

- **Setup:** The challenger runs the setup algorithm to send the public parameter PK to the adversary.
- **Phase 1:** The adversary generates multiple private keys corresponding to various attribute sets S1,…,Sq1.
- **Challenge:** Two identical-sized messages, M_0 and M_1 are submitted by the adversary to the challenger. In addition, a challenging access structure A* that is not satisfied by any of the sets $S_1,\ldots,S_{q1}$ from Phase 1 is presented by the adversary. Next, the coin b is randomly flipped by the challenger to encrypt M_b using $\mathbb{A}$*. Finally, the generated ciphertext CT* is sent to the adversary.

Proceedings DOI: 10.21467/proceedings.7.6
ISBN: 978-81-989164-9-5

Series: AIJR Proceedings
ISSN: 2582-3922

- **Phase 2:** To ensure none of the attribute sets $S_{q1+1},\ldots,S_q$ meets the access structure A* that is associated with the challenge, phase 1 is iterated.
- **Guess:** A guess b' of b is generated by the adversary. If b'=b, the adversary can win the game

The effectiveness of adversary A is measured by $\Pr[b' = b] - ½$. To account for chosen ciphertext attacks, the framework can be extended by integrating decryption queries during Phase 1 and Phase 2. The CP-ABE system is deemed safe when any adversary operating within polynomial time constraints has only a slight gain in the game.

4 Proposed System

The proposed Blockchain-based CP-ABE system as illustrated in Fig. 4 comprises six entities and five algorithms. The entities of the system are:

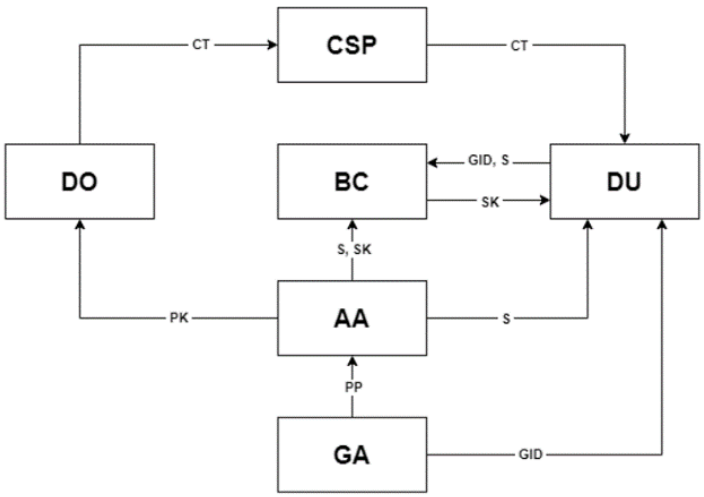

Fig 4. Blockchain-based CP-ABE system model

1) **Global Authority (GA):** The GA creates global public parameters (PP) for the system and Global identity (GID) for all the registered users of the system. It then issues PP to AA and GID to all the registered users of the system. GA is an entirely reliable entity of the system.
2) **Attribute Authority (AA):** The AA issues and revokes the attributes of the users depending on their part in the system. It generates PK and MSK using the user's attribute set. PK is published to all users whereas MSK is kept by itself. AA stores the attribute list of each user along with SK in the blockchain. AA is a partially trusted entity of the system.
3) **Blockchain (BC):** Blockchain is an immutable digital ledger that holds users' metadata. The BC issues the SK of each user after successful verification of attributes of DU.
4) **Cloud Service Provider (CSP):** The CSP provides a data storage solution to DO. Prior to uploading information to the cloud repository, the DO encrypts it using a predetermined access control protocol. This process involves the DO securing their data through encryption based on a specific access management strategy before transferring it to the cloud-based storage system provided by the CSP.
5) **Data Owner (DO):** The DO establishes the attribute-based access control policy within the system. Data is encrypted in accordance with this policy prior to being uploaded to cloud storage. The users who fulfil the requirements set by the access control policy can decrypt the encrypted data.

Proceedings DOI: 10.21467/proceedings.7.6
ISBN: 978-81-989164-9-5

Series: AIJR Proceedings
ISSN: 2582-3922

6) **Data User (DU):** The encrypted information is retrieved by the DU from the CSP. To access the encrypted content, the data user's attributes must match with the access control policy. The DU cannot be considered a fully trustworthy entity within the system.

Initially, the DO encrypts message M using sG, where G represents the generator of an elliptic curve's cyclic subgroup with order r, and s is an arbitrarily selected value from $\mathbb{Z}_r$. Based on LSSS matrix, the encryption process then divides s into shares λx, while 0 is similarly divided into shares ω_x. To decrypt message M, the DU must combine elements of the ciphertext with their attribute keys to obtain the blinding factor sG. The proposed system consists of six distinct algorithms:

1) **Global Setup:** This algorithm is executed by GA. Consider a finite field of order q is GF(q), and E is an elliptic curve defined over GF(q). Consider G be a generator of cyclic subgroups on the elliptic curve E with a large prime order r, where solving the elliptic curve discrete logarithm problem (ECDLP) is difficult. Furthermore, the hash function $H : \{0,1\}^* \rightarrow \mathbb{Z}r^*$ is selected to associate the user's GID to the elements in $\mathbb{Z}_r$. The public parameters of the system where $PP = \{GF(q), G, E, \mathbb{A}, H\}$. The GA forwards PP to AA.
2) **Authority Setup:** The AA executes his algorithm. The AA picks an arbitrary value $n \in \mathbb{Z}r$ its MSK and publishes nG as its corresponding PK. For every attribute i in the system, the AA chooses a random $k_i \in \mathbb{Z}r$ and publishes $PK_i = k_i G$ as a public key. The AA stores the attribute list of each data user in the blockchain.
3) **Encryption:** DO executes this algorithm. The encryption process involves the following stages:

   Step 1: At the outset, the original message is mapped to elliptic curve E at a point M, after which the following computation is performed: $C_0 = M + sG$, where s ∈ $\mathbb{Z}_r$ represents an arbitrary number.

   Step 2: The encryption process utilizes DO's access policy to generate an $n\ x\ \ell$ access matrix A, where ρ maps the rows to the attributes.

   Step 3: Choose two arbitrary vectors $v = \{s, v_2, v_3, \ldots v_m\} \in \mathbb{Z}_r$ and $u = \{0, u_2, u_3, \ldots u_m\} \in \mathbb{Z}_r$ and compute: $\lambda_x = \mathbb{A}_x.v$ and $\omega_x = \mathbb{A}_x.u$, where $\mathbb{A}_x$ represents the row x of $\mathbb{A}$

   Step 4: The two components of ciphertext: C_1 and C_2 are computed as: $C_{1,x} = \lambda_x G + \omega_x PK_{\rho(x)}$ and $C_{2,x} = \omega_x G \quad \forall x$

   Step 5: Using SHA1, message authentication code MAC_m is generated. $MAC_m = SHA1(M)$
4) **Key Generation**: The AA creates the secret key SK associated with attribute i for a user with a GID, and includes this attribute i in its respective attribute list. $SK_{i,GID} = k_i + H(GID)_n$
5) **Decryption:** DU executes this algorithm. The ciphertext is presumed to have been encrypted using an access matrix $(\mathbb{A}, \rho)$. Any user having the secret key $\{SK_{\rho(x)}, GID\}$ for a subset of rows $\mathbb{A}_x$ in $\mathbb{A}$, where $(1, 0, \ldots, 0)$ is contained within the span of these rows. To decrypt the ciphertext, the user needs to determine $H(GID)$ for each x. Subsequently, they should compute:

$$\begin{aligned} \sum C_{1,x} - \sum C_{2,x}\ SK_{\rho(x),GID} &= \sum(\lambda_x G + \omega_x\ PK_{\rho(x)}) - \sum (\omega_x\ G(k\ \rho_{(x)} + H(GID)\ n) \\ &= \sum (\lambda_x G + \omega_x\ k\ \rho_{(x)}\ G) - \sum (\omega_x\ k\ \rho_{(x)}\ G + \omega_x\ H(GID)\ nG) \\ &= \sum \lambda_x G - \omega_x\ H(GID)\ nG \end{aligned}$$

The DO picks constants $c_x \in \mathbb{Z}_r$ such that $\sum_x c_x \mathbb{A}_x = (1,0,\ldots,0)$ $\sum_x c_x \mathbb{A}_x = (1,0,\ldots,0)$ and then perform computation:

$$\sum c_x(\lambda_x G - \omega_x H(GID)nG) = sG \text{ as} v \cdot (1,0,\cdots,0) = s \text{ and } u \cdot (1,0,\cdots,0) = 0.$$

Ultimately, the DU can perform the computation: $C_0 - sG = M$ and then map M back to the original message. Now, calculate the message authentication code ($MAC_{m'}$) of retrieved message

Proceedings DOI: 10.21467/proceedings.7.6
ISBN: 978-81-989164-9-5

Series: AIJR Proceedings
ISSN: 2582-3922

M' using SHA1 hash function: $MAC_{m'} = SHA1(M')$. If, $MAC_m = MAC_{m'}$ then it confirms that the message M has been appropriately received.

6) **Attribute Revocation:** This system enables the revocation of attributes by storing and managing user secret keys on the blockchain through the attribute authority. This method enables the immediate revocation of a user or an attribute without requiring updates to the keys of other users during the revocation. For user revocation, the attribute authority must eliminate the attribute list linked to the user's GID. For attribute revocation, the attribute authority must remove the public key associated with the specific attribute. While revoking the attributes of the user, the attribute authority is required to delete them from the user's attribute list.

5 Security Analysis

The security analysis of the proposed CP-ABE scheme concerning the main security requirements, including privacy, security, confidentiality, integrity, collusion resistance, and forward security is evaluated in this section.

5.1 Data Security

Our system employs an ECC algorithm grounded in ECDLP to prevent unauthorized access to the secret key without proper attributes. The encryption of message M using ECC results in ciphertext C_0. If M can be represented as mG, where m is an element of $\mathbb{Z}_r$ and s is a random value chosen by the data owner, then $C_0 = (m + s)G$ appears as an arbitrary point on the elliptic curve to potential attackers. Due to the properties of ECDLP, attackers are unable to extract any meaningful information about M without knowledge of s. LSSS divides the secret s into λx shares, that can only be restored when an adequate quantity of shares is present. This means that decryption of the ciphertext is possible if the data user pleases the access structure $(\mathbb{A}, \varrho)$ and holds the necessary attributes. Unauthorized users lacking the specified attributes will not have corresponding attributes in rows $\mathbb{A}$ x, preventing them from calculating $\sum_x c_x \lambda_x = (1,0,\cdots,0)$. Consequently, they cannot compute the first element s in vector v. This mechanism ensures the security of data in the proposed system.

5.2 Data Privacy and Confidentiality

Our system restricts access to secret key SK on the blockchain to users with appropriate credentials. Every blockchain transaction is safeguarded using cryptographic methods to ensure data security and privacy. The system employs ECC technology, which relies on the computational complexity of solving the ECDLP problem. Consequently, users lacking the necessary attributes cannot derive the secret key from the public key. This approach effectively safeguards data privacy and confidentiality within our system.

5.3 Data Integrity

The data integrity is assured by the proposed system through the utilization of MAC. It verifies MAC_m of the original message with $MAC_{m'}$ of the received message i.e, whether $MAC_m = MAC_{m'}$. If the answer is Yes, then the integrity of the message is preserved, otherwise it is not.

5.4 Collusion Resistance

The attribute of each user is associated with a GID to provide a safeguard against collusion attacks. This ensures that different attributes of a user cannot be effectively combined during decryption. In the event that two users with separate identities attempt to collude, certain elements like $H(GID)\omega_x nG$ will remain, making it impossible to eliminate them. Consequently, this attempt will fail to recover sG and message M. For example [18], Consider, Alice possesses two attributes A and B, Bob possesses three attributes C, D, and E. and the access policy is $A \wedge ((B \vee C) \wedge (D \vee E))$. Both Alice and Bob cannot decrypt the ciphertext independently. If they collude their attributes to decrypt a ciphertext, they will obtain the distinct

Proceedings DOI: 10.21467/proceedings.7.6
ISBN: 978-81-989164-9-5

Series: AIJR Proceedings
ISSN: 2582-3922

result for some *x*. Alice will obtain: $\lambda_x G - H(GID_{Alice})\,\omega_x nG$ and Bob will obtain: $\lambda_x G - H(GID_{Bob})\,\omega_x\,nG$. Alice and Bob had distinct GIDs, therefore, $H(GID_{Alice}) \neq H(GID_{Bob})$ makes $\sum_x H(GID)\,\omega_x\,nG \neq 0$. Usually, authorized users choose constants $c_x \in \mathbb{Z}_r$ and $\sum_x c_x \mathbb{A}_x = (1,0,\cdots,0)$ to determine sG. Alice and Bob are unable to collude to obtain constants $c_x \in \mathbb{Z}_r$ and $\sum_x c_x \mathbb{A}_x = (1,0,\cdots,0)$ to retrieve sG. Therefore, our scheme is collusion-resistant.

5.5 Forward Security

Our system enforces forward security, ensuring that revoked users cannot access future data. This protection is achieved by the attribute authority simply removing the attribute list linked to a user's GID when revoking access. If a revoked user tries to decrypt, their request is rejected since their GID is unrecognized in the system, and the authority cannot confirm their attribute ownership. Additionally, the authority can modify the attribute list to revoke a specific user attribute. Decryption requests from users lacking the necessary attributes are rejected. This approach effectively ensures forward security for outsourced data, safeguarding it from unauthorized access by revoked users.

6 Conclusion

This study introduces a new approach to access control employing blockchain and CP-ABE. The proposed method involves encrypting data prior to its transfer to cloud storage, with the decryption key stored on the blockchain. Access to this private key on the blockchain is restricted to users possessing the necessary attributes, making sure that only permitted users are able to decrypt and access the saved data. The use LSSS access structure can increase the intricacy of the access framework, thereby enhancing data protection. Each transaction on the blockchain is protected using cryptographic principles to maintain data privacy and confidentiality. The system ensures the integrity of the message through MAC. A unique GID is associated with every user attribute to safeguard against collusion attacks. The revocation of the user or user attribute ensures the forward security data. The overall system can enhance the privacy, security, confidentiality, integrity, and forward security of the data and made the system collusion-resistant.

7 Competing Interests

No potential conflict of interest exists in this publication.

References

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," *Future Generation Computer Systems*, vol. 25, no. 6, p. 17, 2011, doi: 10.1109/EIDWT.2013.106.

[2] A. Sahai and B. Waters, "LNCS 3494 - Fuzzy Identity-Based Encryption," 2005.

[3] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data," 2006.

[4] J. Bethencourt, A. S. Ucla, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," 2007.

[5] M. Horváth, "Attribute-Based Encryption Optimized for Cloud Computing."

[6] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain Technology Overview," 2018. doi: 10.6028/NIST.IR.8202.

[7] S. Singh, Y. S. Jeong, and J. H. Park, "A survey on cloud computing security: Issues, threats, and solutions," *Journal of Network and Computer Applications*, vol. 75, pp. 200–222, Nov. 2016, doi: 10.1016/j.jnca.2016.09.002.

[8] P. Yang, N. Xiong, and J. Ren, "Data Security and Privacy Protection for Cloud Storage: A Survey," *IEEE Access*, vol. 8, pp. 131723–131740, 2020, doi: 10.1109/ACCESS.2020.3009876.

[9] P. J. Sun, "Security and privacy protection in cloud computing: Discussions and challenges," *Journal of Network and Computer Applications*, vol. 160, no. August 2019, p. 102642, 2020, doi: 10.1016/j.jnca.2020.102642.

[10] P. Sharma, R. Jindal, and M. D. Borah, "Blockchain Technology for Cloud Storage: A Systematic Literature Review," Sep. 01, 2020, *Association for Computing Machinery*. doi: 10.1145/3403954.

[11] J. H. Park and J. H. Park, "Blockchain security in cloud computing: Use cases, challenges, and solutions," *Symmetry (Basel)*, vol. 9, no. 8, pp. 1–13, 2017, doi: 10.3390/sym9080164.

[12] J. Li, J. Wu, and L. Chen, "Block-secure: Blockchain based scheme for secure P2P cloud storage," *Inf Sci (N Y)*, vol. 465, pp. 219–231, Oct. 2018, doi: 10.1016/j.ins.2018.06.071.

[13] M. A. Darwish, E. Yafi, M. A. Al Ghamdi, and A. Almasri, "Decentralizing Privacy Implementation at Cloud Storage Using Blockchain-Based Hybrid Algorithm," *Arab J Sci Eng*, vol. 45, no. 4, pp. 3369–3378, Apr. 2020, doi: 10.1007/s13369-020-04394-w.

[14] G. Lin, H. Hong, and Z. Sun, "A Collaborative Key Management Protocol in Ciphertext Policy Attribute-Based Encryption for Cloud Data Sharing," *IEEE Access*, vol. 5, pp. 9464–9475, 2017, doi: 10.1109/ACCESS.2017.2707126.

[15] V. Odelu and A. K. Das, "Design of a new CP-ABE with constant-size secret keys for lightweight devices using elliptic curve cryptography," *Security and Communication Networks*, vol. 9, no. 17, pp. 4048–4059, Nov. 2016, doi: 10.1002/sec.1587.

[16] S. Wang, K. Liang, J. K. Liu, J. Chen, J. Yu, and W. Xie, "Attribute-Based Data Sharing Scheme Revisited in Cloud Computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1661–1673, Aug. 2016, doi: 10.1109/TIFS.2016.2549004.

[17] R. M. Basavarajegowda and S. M. Sundaram, "Enhanced CP-ABE with RSA for Secure and Revocable Data Transmission of Big Data in Cloud," *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 2, pp. 47–56, Apr. 2022, doi: 10.22266/ijies2022.0430.05.

[18] S. Ding, C. Li, and H. Li, "A Novel Efficient Pairing-Free CP-ABE Based on Elliptic Curve Cryptography for IoT," *IEEE Access*, vol. 6, pp. 27336–27345, May 2018, doi: 10.1109/ACCESS.2018.2836350.

[19] R. Cheng, K. Wu, Y. Su, W. Li, W. Cui, and J. Tong, "An efficient ECC-based cp-ABE scheme for power IOT," *Processes*, vol. 9, no. 7, Jul. 2021, doi: 10.3390/pr9071176.

[20] P. Sharma, R. Jindal, and M. D. Borah, "Blockchain-based decentralized architecture for cloud storage system," *Journal of Information Security and Applications*, vol. 62, Nov. 2021, doi: 10.1016/j.jisa.2021.102970.

[21] Y. Zhang, X. Wei, J. Cao, J. Ning, Z. Ying, and D. Zheng, "Blockchain-Enabled decentralized Attribute-Based access control with policy hiding for smart healthcare," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, pp. 8350–8361, Nov. 2022, doi: 10.1016/j.jksuci.2022.08.015.

[22] X. Yang and C. Zhang, "Blockchain-Based Multiple Authorities Attribute-Based Encryption for EHR Access Control Scheme," *Applied Sciences (Switzerland)*, vol. 12, no. 21, Nov. 2022, doi: 10.3390/app122110812.

[23] L. Hong, K. Zhang, J. Gong, and H. Qian, "A Practical and Efficient Blockchain-Assisted Attribute-Based Encryption Scheme for Access Control and Data Sharing," *Security and Communication Networks*, vol. 2022, 2022, doi: 10.1155/2022/4978802.

[24] G. Sucharitha, V. Sitharamulu, S. N. Mohanty, A. Matta, and D. Jose, "Enhancing Secure Communication in the Cloud Through Blockchain Assisted-CP-DABE," *IEEE Access*, vol. 11, pp. 99005–99015, 2023, doi: 10.1109/ACCESS.2023.3312609.

[25] U. Bodkhe *et al.*, "Blockchain for Industry 4.0: A comprehensive review," *IEEE Access*, vol. 8, pp. 79764–79800, 2020, doi: 10.1109/ACCESS.2020.2988579.

[26] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain Technology Overview Blockchain," 2018.

[27] Y. Wang, L. Wei, X. Tong, X. Zhao, and M. Li, "CP-ABE based access control for cloud storage," in *Advances in Intelligent Systems and Computing*, Springer Verlag, 2017, pp. 463–472. doi: 10.1007/978-3-319-38771-0_45.

[28] A. Beimel, "Secure Schemes for Secret Sharing and Key Distribution," 1996. Accessed: Feb. 04, 2024. [Online]. Available: https://www.cs.bgu.ac.il/~beimel/Papers/thesis.pdf

[29] A. Lewko and B. Waters, "Decentralizing Attribute-Based Encryption," in *K.G. Paterson (Ed.): Eurocrypt 2011, LNCS 6632, pp. 568–588*, 2011. Accessed: Feb. 04, 2024. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-20465-4_31

[30] Z. Liu, Z. Cao, D. S. Wong, and Z. Liu, "Efficient Generation of Linear Secret Sharing Scheme Matrices from Threshold Access Trees", [Online]. Available: https://eprint.iacr.org/2010/374.pdf