Hindawi
Security and Communication Networks
Volume 2022, Article ID 8685273, 11 pages
https://doi.org/10.1155/2022/8685273

Research Article

An Efficient Blockchain Based Data Access with Modified Hierarchical Attribute Access Structure with CP-ABE Using ECC Scheme for Patient Health Record

F. Sammy[1] **and S. Maria Celestin Vigila**[2]

[1]*Department of Information Technology, Dambi Dollo University, Dembi Dolo, Welega, Ethiopia*
[2]*Department of Information Technology, Noorul Islam Centre for Higher Education, Kumaracoil, Tamil Nadu, India*

Correspondence should be addressed to F. Sammy; sammy@dadu.edu.et

Received 21 January 2022; Revised 2 February 2022; Accepted 12 February 2022; Published 8 March 2022

Academic Editor: G. Thippa Reddy

Secure patient health record (PHR) information exchange via cloud computing is a considerable security risk to user privacy. The fundamental reason of this issue is cloud computing's reliance on trustworthy third parties to share data across it. To exchange data securely, many conventional cryptographic algorithms employ various keying approaches. However, relying on a trusted third party compromises the privacy of consumers' data. To offer secure communication without the involvement of a third party, a distributed blockchain based (DBC) ciphertext–policy attribute-based encryption (CP-ABE) approach is employed in this study. Because of bilinear paring and simple scalar multiplication factors, the proposed CP-ABE system is entirely dependent on elliptic curve cryptography to reduce complexity. Furthermore, the data requester provides dynamic attributes, and a user-centric access policy is created, allowing multiple authorities to manage the attributes and provide data access. Data confidentiality, data authentication, user authentication, and tamper-proof data are all guaranteed by the suggested method. The DBC-CP-ABE method is used to provide user-centric access policies and effective key management.

1. Introduction

The Internet of Things (IoT) is a new technology that allows items to communicate with one another across wireless networks. IoT devices are resource-constrained and have challenges with data processing, data storage, and energy consumption. Cloud computing provides a centralized solution to these resource restricted procedures to overcome these limits. The collected data is stored and processed in the cloud, but the cloud can be a trustless environment with major security issues such as single point failure, data tampering, lack of user privacy due to a lack of data access control, Denial of Service (DoS), Man-in-the-Middle attack (MiTM), and password phishing. As a result of permitted data/device access, the cloud environment is prone to security breaches, compromising users' privacy. Many public key secret writing techniques provide a fine-grained access control strategy while also protecting the privacy of users. Among other public key encryption methods, CP-ABE scheme offers one-to-many access control which allows data to be shared across multiple users. But the implementation overhead incurs due to operations with bilinear pairing. It consumes more resources with high computational cost. To overcome this issue, less complex and less resource consumption scalar computation with elliptic curve cryptography (ECC) is used in this work. This reduces the computational requirement by two-three times that of bilinear pairing. This work focuses on building a security system with blockchain where hierarchical access control policy is achieved by combining CP-ABE and ECC. The experiment analysis shows that our scheme outperforms the compared work in cryptographic operations. The major focus of this work is concentrated to achieve the following criteria:

(1) Adoption of straightforward scalar multiplication with ECC and CP-ABE approach reduces procedure overhead caused by bilinear pairing methodology

(2) The proposed method ensures use of multiple authorities to manage attributes and shares multiple data attributes of single data user

(3) To specify the access policy scheme with increased security, the Linear Secret Sharing Scheme (LSSS) is used

(4) Attribute revocation for a data user is achieved with the help of RSA key pair in communication between the data user and Attribute Authority (AA)

The following are the last sections: Section 2 contains material from the research study that is relevant to the current effort, and Section 3 contains information on the proposed study's contribution. The preliminaries utilized on this project are explained in Section 4. The architecture of the blockchain based hierarchical access control scheme with CP-ABE using ECC is briefly described in Section 5. The modified hierarchical attribute access structure (MHAAS) with CP-ABE employing ECC is explained in Section 6. The integration of HACS-CP-ABE-ECC with blockchain is explained in Section 7. Section 8 summarizes the HACS-CP-ABE-ECC with blockchain security analysis, whereas Section 9 describes the performance evaluation conducted in this study.

2. Related Works

Cloud computing offers computation of massive data and data sharing in a promising way [1]. Data are encrypted and shared in cloud computing environment either with symmetric key encryption or public key standards [2–4]. This method has drawback in achieving security [2] and drawback in flexible access control [3] and shows poor performance [4]. To deal with these drawbacks, attribute-based encryption (ABE) is proposed. There are two types of attribute-based encryption: KP-ABE and CP-ABE. Bethencourt et al. [5] were the first to suggest CP-ABE. ABE scheme with bilinear pairing showing less efficiency was proposed [6]. ABE is further refined with CP-ABE involving hierarchical attributes as proposed by [7] to address key management problem [8]. A multiauthority-ABE with dynamic policy attributes is proposed, although the CP-ABE method demonstrates little improvement [9]. An access policy based on the DBDH scheme is proposed. All the CP-ABE methods described above use bilinear mapping using large sized keys. To lessen the complexity of CP-ABE, the decryption method is split into degrees: predecryption and final decryption degrees in [10]. But this method does not ensure forward security. This has been improved in other work [11] where encryption and decryption are outsourced and validated but lack improvement in encryption and decryption process. The work is also extended in [12] by redistributing the encoding and decoding system to fog nodes; however, they are easily attacked. Another decryption outsourcing work proposed in [13] resists against selective ciphertext. Although the work in [14] provides outsourcing of encryption and decryption process, it uses bilinear pairing that remains as hurdle to performance improvement to CP-ABE.

CP-ABE does not ensure less storage overhead and good cost-effective solution as it depends on the use of bilinear maps. A bilinear map produces secret keys of larger values and ciphertext with linear associated attribute. And also it uses exponentiation factors for doing encryption and decryption process which relies on linear attributes defined in the access policy [15–17]. The problem of requiring a large key size necessitates the usage of elliptic curve cryptography (ECC) with a smaller key size. This paves the path for CP-ABE to define an access structure utilizing ECC [18–20]. Lightweight devices such as the CP-ABE with constant key size using ECC have been developed, but they are not appropriate for complex access structures [21, 22]. Another lightweight work using KP-ABE without bilinear pairing is proposed but suffers from poor scalability and lack of decryption outsourcing [23]. The overall computation overhead due to bilinear pairing is overcome with ECC [24]. Constant key size with CP-ABE using ABE addressing collision attack problem is proposed in [25]. Alternative to bilinear pairing with ECC to address secured data share is proposed in [26]. All the abovementioned work defines the access policy based on the set of attributes.

Betherncourt created the first tree-based access control structure in order to implement AND, OR, and OF strategies [27]; however, it is insecure. Many studies focus on improving access control strategies; however, the time it takes to encrypt and decrypt data grows as the number of attributes increases. The research was furthered by Lewko and Waters, who proposed a technique to convert tree access control to an LSSS and Waters enhanced CP-ABE with a matrix format [28]. With d-parallel BDHE assumption, this gives security. Many studies have refined the use of CP-ABE with flat access control [29–33], constant ciphertext [34], accountability and authorities with attribute revocation [35–38], and improvement in security through accountability and authorities. However, none of these structures support hierarchical file relationships.

Hierarchical CP-ABE based on LSSS matrix structure was also studied. By considering hierarchical heads sharing secret keys with users, these approaches lessen the burden of a single head [39, 40]. In this paper, we design a hierarchical based access relation for sharing multiple files [41] in a distributed blockchain context using LSSS. To address the privacy and security concerns, [45] present a unique pairing-free certificateless method that builds a novel reliable and efficient lightweight certificateless signature (CLS) scheme using a state-of-the-art blockchain technique and smart contract. Paper [46] addresses a lightweight and reliable authentication protocol for wireless medical sensor networks (WMSN), which is composed of cutting-edge blockchain technology and physically unclonable functions (PUF), to address physical layer security and the over-centralized server problem in WMSN. The elliptic curve digital signature algorithm (ECDSA), which is one of the essential building blocks of blockchain, is proposed in [47] as an efficient and large-scale batch verification technique with

group testing technology. Using edge computing and blockchain approaches, [48] introduces search efficiency, reliability requirements, and a resource allocation scheme to properly handle IoT devices. The study [49] demonstrates how to use erasure coding to overcome data integrity issues in IoT devices.

3. Our Contribution

We suggested a blockchain based hierarchical access scheme that uses CP-ABE with ECC in this paper. A hierarchical access hierarchy is defined here, with the user attribute satisfying partially or entirely alone allowing partial or complete access to the data. A root authority (RA) checks and joins all of the domain attribute authorities (AA) in the blockchain. For each AA, RA produces a public key and a master key. It also sends hierarchical access scheme to all AA. RA sends the public key to AA while keeping the master key hidden. AA takes an attribute from the users and generates an address, an RSA key pair, and a private key for that attribute. Based on this, AA distributes the attribute's address, RSA key pair, and private key to the user who satisfies the access structure to decrypt the data. The AA keeps track of the RSA key pair in order to revoke the user's attribute. To reduce computing complexity, the pre-decryption is outsourced to AA, and AA's trust is kept thanks to the presence of blockchain. The suggested method ensures that data is shared with several authorities and that different attributes of the user's identification are shared.

4. Preliminaries

4.1. Elliptic Curve Cryptography. ECC is a discrete logarithm problem-based public key cryptography (ECDLP). The elliptic curve E is defined by FG (P), a finite field, and is written as $y^2 = x^3 + ax + b \pmod{p}$ and $4a^3 + 27b^2 \neq 0$. Calculate a point on the curve $Q = KG$, where G is the prime order r generator group over the polynomial time k. The plain texts are transferred to the elliptic curve's point Q. The ECC procedure is broken down into three phases.

(a) Key generation:

(1) Both the data server and the data client have agreed to use the same elliptic curve $y2 = x3 + ax + b \pmod{p}$ and G

(2) The data server generates a random number, $Sa \in Zp$, as the private key, and $Pa = SaG$, as the public key

(3) Data clients generate a private key using a random number $Sb \in Zp$ and a public key using $Pb = SbG$

(b) Encryption:

The data server encrypts the message with Q by selecting a random number $K \in Zp$, then computes the cypher text $C1 = KG$ and $C2 = Q + K\ Pb$, and sends both C1 and C2 to the data clients

(c) Decryption:

Data clients use $C_2 - S_bC_1 = Q + kP_b - S_bkG = Q$ to decrypt the message. The message is obtained by mapping to the curve's point Q.

4.2. Hierarchical Access Control Strategies. As demonstrated in Figures 1 and 2, a hierarchical access control technique allows numerous access structures to be combined into a single structure T.

4.3. Linear Secret Sharing Scheme. Beimel proposed the Linear Secret Sharing Scheme [33]. When all parties make up a share on vector Zp, a Secret Sharing Scheme is defined across linear Zp for various parties. Matrix M was created to generate shares for all parties. Consider the M matrix, which has p rows and q columns. Consider a row of a matrix Mi where i= (1, 2, ..., p) meets the criterion 1, 2, ..., p) $\longrightarrow d$, and given a column vector $\bar{O}$ = (s, u2...un) with the secret key $s \in$ Zp and u2. . ..un $\in\in$ Zp picked at random. M is made up of m shares of s, each of which is dependent. The share mi = (M $\bar{O}$)i belongs to a specific political party.

Consider an LSSS Π with T as the access tree structure and S $\in T$. This denotes an arbitrary permitted set, $L \subset$ {1 ...p}, and $L = \{i$: mi $\in S\}$.$s = \sum_{i \in L} \omega_{i m_i}$ and m_i are arbitrary secret s specified by ω_i which is discovered in the matrix M in polynomial time. There is a vector $\omega.(1, 0, \ldots..0) = -1$ and. $\omega.M_i = 0$ for the unlawful set of rows $i \in L$.

When a jth secret of a nonleaf node is recovered from a set of n secrets, the set of attributes $\{\omega_{i \in Zp}\}$ can be discovered in polynomial time by satisfying $\sum_{i \in L} \omega_{i,j} M_i^T = \varepsilon_j$, where j denotes a row vector of length n with the jth element equal to 1 and all other elements equal to 0. As a result, secret share $s_j = \sum_{i \in L} \omega_{i,j} m_i$. The marking method defined by [34] is used to create the LSSS matrix. It translates a Boolean formula-defined access tree to the LSSS matrix technique. In hierarchical access control, this LSSS marking mechanism is employed. According to Figure 3, if the user characteristics only partially satisfy the access structure policy, just a portion of the information is decrypted.

5. Architecture of Blockchain Based Hierarchical Access Control Scheme with CP-ABE Using ECC (BHACS-CP-ABE-ECC)

Certificate authority (CA), attribute authority for personal, health, and insurance domain, cloud service provider, data owner (DO), data clients (DC), and edge nodes for pre-decryption process are all part of the proposed blockchain linked architecture. Figure 3 depicts the proposed scheme framework. The following is a description of each participant's functionality:

(1) Root authority (RA): the main role RA is to provide identity of the communicating nodes by considering the security parameter (K) of the node and generates public parameter (PP). To generate a public key and master key for an attribute, this public parameter is submitted to the appropriate attribute authority.

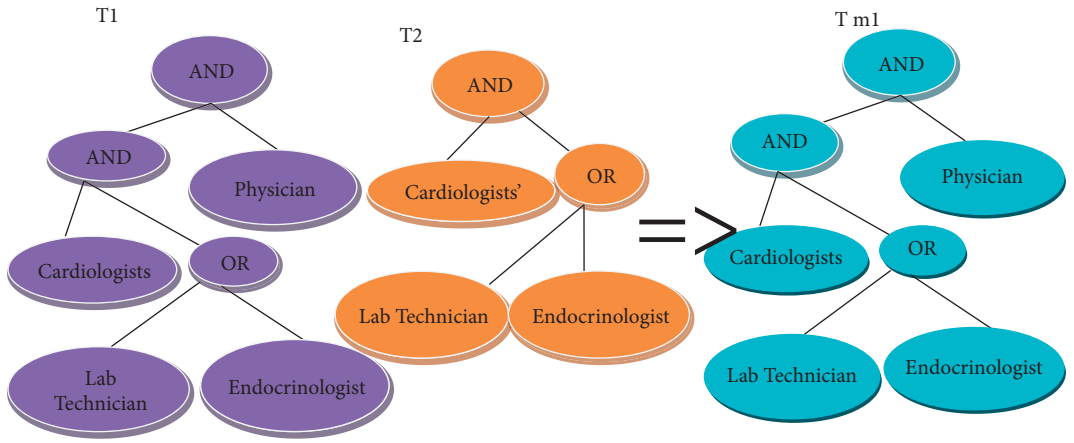

Figure 1: Part of integrated hierarchical access control structure.

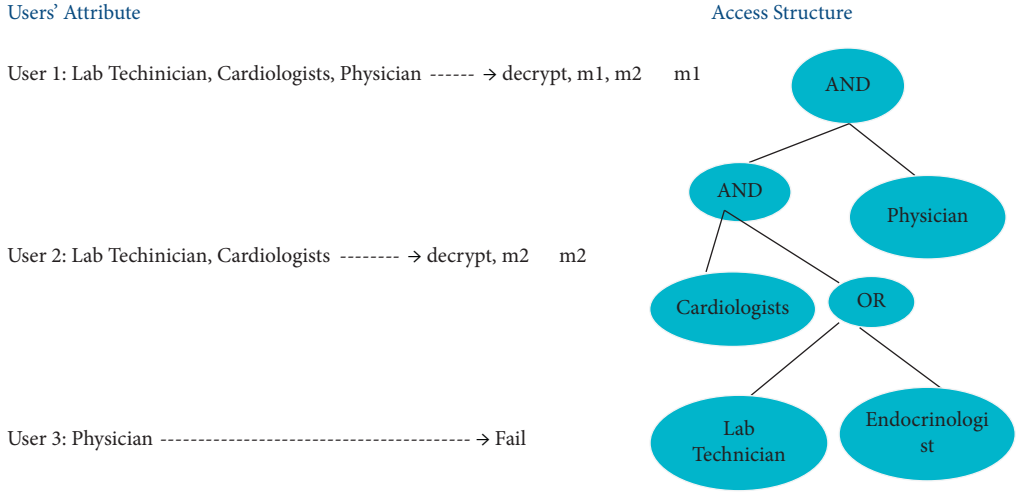

Figure 2: Part of integrated hierarchical access control process.

(2) Attribute authority (AA): attribute authority of the domain extracts the attributes of their respective data clients. Attribute authority further generates public key and master key for that attribute.

(3) Cloud storage: cloud storage serves to store encrypted data (CT) sent by the data owners.

(4) Data owners (DO): the data is encrypted before being uploaded to the cloud server by the data owners. It creates ciphertext CT using plaintext B, the matching public key PK, and the access policy given by the LSSS matrix structure (M,m).

(5) Data clients (DC): data clients are responsible for performing decryption on CT. Deciphering is done in two stages. First the local server near the DC serves as edge nodes and does partial encryption by inputting CT and SK. Finally the DC decrypt the partial decrypted CT to plaintext by considering CT′ and DSK.

6. Modified Hierarchical Attribute Access Structure (MHAAS) with CP-ABE Using ECC

The following section explains the process carried out using hierarchical access policy structure (Schemes 1–5).

7. Integration of HACS-CP-ABE-ECC with Blockchain

The hierarchical access control scheme employing with ciphertext ABE using ECC is integrated with blockchain and its operation is explained below.

The operation of this method is explained as six principal components as initialization phase, authority creation, user creation, ciphertext data upload, creation and issuance of attributes, and revoke attribute. This process includes the reception of only the secret key of the attribute for a particular address in its wallet alone is specified in the process.

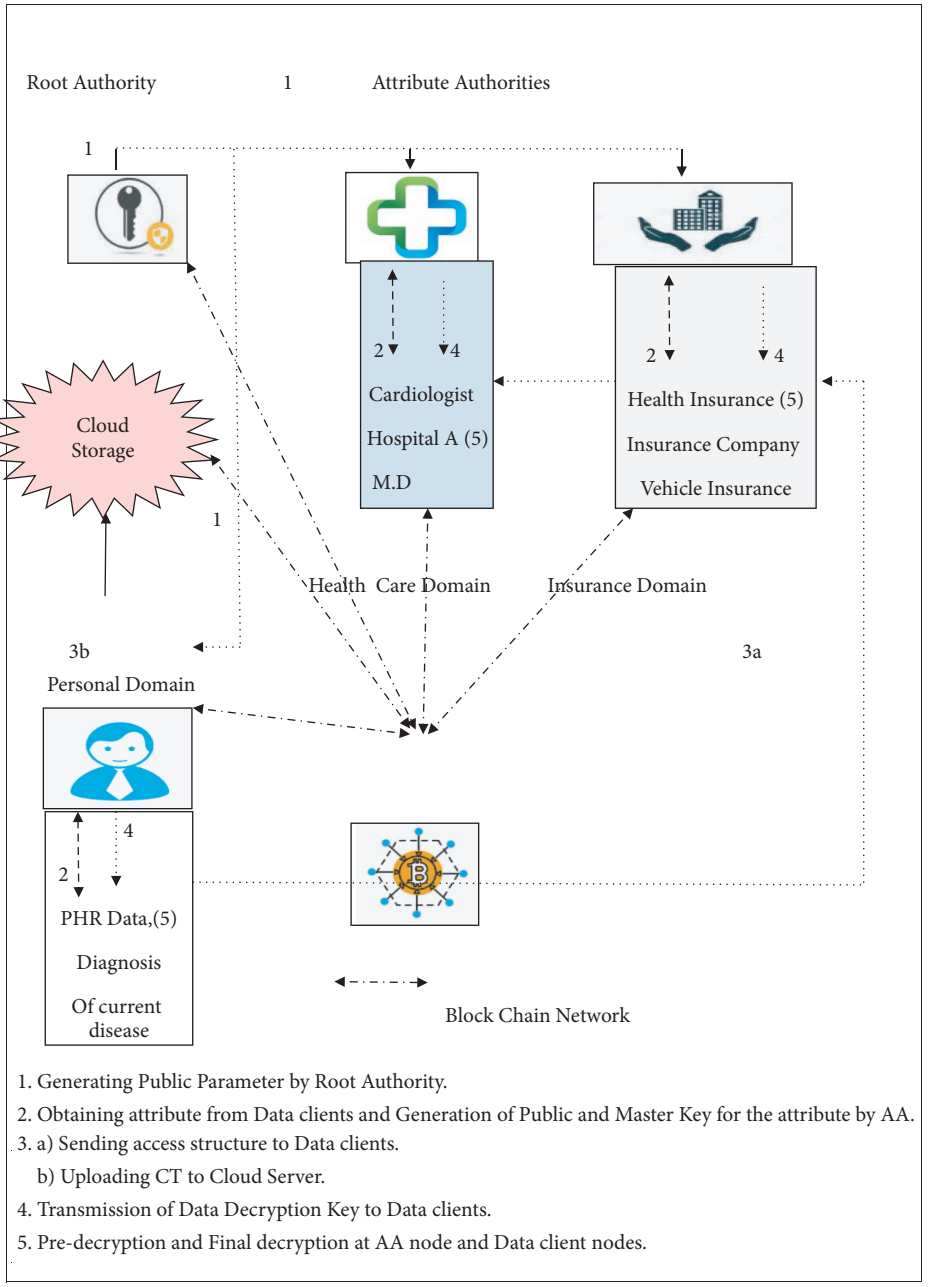

Figure 3: Architecture of proposed blockchain based patient centric data access with CP-ABE using ECC.

Input: Security parameter K.

Output: Public parameter PP.

1. Select G Generator of cyclic subgroups with prime order r on E, GF (q) of order r, Elliptic Curve E defined over GF (q).
2. Select H: 0, 1* Z_r as the hash function, and map the elements of Z_r to the users' GIDs.
3. Define the global attributes $A = \{a_1, a_2, \ldots a_n\}$. Each attribute is defined in the form of LSSS Matrix M_i. representing hierarchical access structure. These attributes are maintained by multiple authorities who generate the necessary key corresponding to the attributes.
4. PP is calculated from the set of $\{GF(q), E, G, h, A\}$.
5. PP is sent to Attribute Authority for generating public and Master key for an attribute.

Scheme 1: System initialization at root authority.

Input: Public parameter *PP*.

Output: Pubic Key *PK* and Master Key *MSK for attribute i*.

1. Select two random numbers $\alpha_i, \beta_i \in Z_r$
2. Generate Master Key = $\{\alpha_i, \beta_i, \forall_i\}$.
3. Generate Public Key = $\{G\alpha_i, G\beta_i, \forall_i\}$.

Scheme 2: Generation of public and master key by attribute authority.

Input: Plaintext set $\{B_j, j \in (p, q)\}$, *PK*, *LSSS* Matrix Structure (M, m)

Output: Cipher Text *CT*.

1. The data owners encrypt the plaintext message B using symmetric key c r and generate the cipher text data $CT_d = E_{c_r}(B)$. using the symmetric key encryption algorithm E (B).
2. Calculates the hash value on the cipher text data $H_{CT}=H(CT_d)\ G$. This ensures data integrity.
3. Data Owners defines LSSS structure (M, m) and sends to the data clients.
4. The encryption algorithm is divided into two stages

   a. Calculate $C_0 = c_r + sG$ *where* $s \in Z_p$.

   b. Select two random vectors $\overrightarrow{v}=(s, v_2, v_3\ldots\ldots, v_m) \in Z_p$ *and* $\overrightarrow{u}=(0, u_2, u_3\ldots\ldots, u_m) \in Z_p$ and calculate $\lambda_x = M_x \cdot \overrightarrow{v}$, $\omega_x = M_x \cdot \overrightarrow{u}$, $C_{1,x} = \lambda_x G + \gamma_x y_{m(x)}\ G$, $C_{2,x} = \gamma_x G$, $C_{3,x} = \omega_x G + \gamma_x k_{m(x)}\ G$ if $m(x) \in$ normal attributes where γ_x *random number* $\in$ *and* $x \in [1, p$ *and* M_x *is the row* x *of* M.

5. Finally calculate $CT = \{(M, m), C_0, \{C_{1,x}, C_{2,x}, C_{3,x}\}, CT_d, H_{CT}\}$. Data Owners uploads *CT* to the Cloud Server Storage.

Scheme 3: Encryption by data owners.

Input: PP, *GID, MSK*

Output: Generation of Private Key for the Data clients .

1. Generates Key $USR_{i,GID'} = y_i + H(GID)\ k_i$ for the user with GID of *i th* attribute.
2. Attribute Authority generates a conversion key $USR_{EN,GID'} = USR_{i,GID}, i \in s_{i,GID}$. It Sends this key to partial decrypting node and Data client node.
3. Data client obtains the private key by calculating $USR_{i,GID} = y_i + H(GID)\ k_i + z$ where z is a random number $\in Z_r$.

Scheme 4: Key generation for data decryption by attribute authority.

The encryption, decryption, and retrieval of public key of the attribute are omitted here.

(a) Initialization phase: this phase includes initialization of blockchain and setting of hierarchical access based policy scheme to all attribute authorities in different domains and provides permission chain through RA. The RA performs mining of the genesis block. For all attribute authority domains, RA is in charge of producing pubic key PK and master key MSK. Everyone has access to the public key, while the master key is kept hidden in order to generate the private key for data decryption.

Input: Cipher Text CT,

Output: Plain Text $\{B_j, j \in (p, q)\}$

Pre-Decryption Stage:

1. For attribute m_x the decryption is obtained as $D_x = C_{1,x-} Sk_{m(x),GID}\, C_{2,x+}\, H(GID)\, C_{3,x}$. At the end of this stage $CT' = \{C_0, CT_d, H_{CT}, T_1, T_2\}$ is sent to the final decryption stage where $T_1 = \Sigma_{x\epsilon X}\, C_x\, D_x = sG - z\, \Sigma_{x\epsilon X}\, C_x\, \gamma_x\, G$ and $T_2 = \Sigma_{x\epsilon X}\, C_x\, C_{2,x} = \Sigma_{x\epsilon X}\, C_x\, \gamma_x G$

Final Decryption Stage:

2. Data Clients calculates $H'_{CT} = H(E_{c_r}(B))\, G$ by using $Cr' = C_{0-} T_1 + zT_2$. If $H_{CT} = H'_{CT}$, the plaintext is valid.

Scheme 5: Decryption by data clients.

(b) Authority creation: RA uses MSK of the Attribute Authority Domains and generates private key SK_{AU} for each AA domain. For this, RA generates new address {p, RSA_{SK}, RSA_{PK}} for each AA domain and transmits public key and RSA key pair to all AA domains. RSA key pair is used to transmit the symmetric encryption key securely to all the requester. RA provides "transmit", "receive," and permission rights to all AA domain and joins the blockchain system.

(c) User creation: now the respective AA adds the individual users and obtains the attributes from the users in the domain. AA generates the address u, RSA key pair, and private key of the individual users SK_{USER}. AA transmits the address u and RSA key pair to the individual users and keeps the SK_{USER} with AA.

(d) Ciphertext data upload: data owners create ciphertext of the data and uploads the hashed ciphertext to the cloud storage.

(e) Attribute creation and distribution: data owners specify the LSSS policy for all data requesters via AA. AA holds the LSSS policy attributes and generates the data decryption private key for the each data requester. Data clients who satisfy the partial or full policy structure are granted access to the data; otherwise, access is refused.

(f) Revoke attribute: since AA holds the attributes of the data clients, it can also withdraw the attributes as it holds the RSA key pair of the attribute for the particular data client. It revokes all the attributes of the data client.

7.1. Functions of the Blockchain Components

7.1.1. Root Authority

BHACS-CP-ABE-ECC:

(1) Initialization phase-(PP⟶PK,MSK)
(2) Authority Creation(PK,MSK,P)⟶ SK_AA1,SK_AA2. . .SK_AAn

Functions of RA:

(1) Blockchain creation
(2) Permits all AA to join the blockchain
(3) Grants connect, send, and receive permission to all AAs

7.1.2. Attribute Authority

(1) User creation
(2) Obtains attributes from the data requester
(3) Creates address and RSA key pair and private key for a particular attribute of the data requester
(4) Sends LSSS access policy to all the data requester
(5) Pre-Decryption of CT
(6) Revoke attributes for a particular data requester

7.1.3. Data Owner

(1) Define and send LSSS access policy to all AA
(2) Data is encrypted and sent to cloud storage using the symmetric encryption technique
(3) Shares the symmetric key securely using RSA key pair to the entire AA

7.1.4. Data Clients

(1) Sends the attribute list to the AA of that domain
(2) Performs final decryption of the data requested

8. Security Analysis of BHACS-CP-ABE-ECC

The following section examines the proposed scheme's security. Under the assumption of DDH, the security model is considered as being secure.

The proposed method supports multiauthority and multiattribute from single data user. For each attribute, attribute authority generates a set of {address, RSA key pair, and private key). The address, master key, and RSA key pair are preserved by the AA to secure the system against adversary attacks and to perform attribute revocation. Instead

of bilinear pairing, the computing phase uses basic scalar multiplication, which makes the procedure more efficient with the compared models. Also, the decryption process is done at two stages: one at the AA end and the other at the data requester which makes the decryption process lighter at the data requester end.

8.1. Security Assumption under Decisional Diffie–Hellman. The proposed model considers Decisional Diffie–Hellman (d-DDH) Assumption and described as follows: the challenger chooses F, a cyclic group with prime order s, and G, a cyclic group F generator, while y and k are chosen at random from Zs. Despite being given a tuple of (G, yG, kG), the adversary finds it difficult to validate y, k, G in polynomial time from random element $X \in$ F. The algorithm A overcomes the DDH problem with a constant factor ρ which is obtained from $|\mathrm{Fs}[\mathrm{A}(G, \mathrm{yG}, \mathrm{kG}, Z = \mathrm{ykG}) = 0] - \mathrm{Fs}[\mathrm{A}(G, \mathrm{yG}, \mathrm{kG}, Z = X) = 0]| \geq \rho$.

8.2. Security under Chosen Ciphertext Attack. The communication between adversary δ and the challenger $\complement$ is given below. The summons is given an access structure (T, m) by the opponent. The initialization algorithm is run by challenger. The system's public parameter is used to compute the public and secret keys, with the public key being sent to the opponent. Stage 1: the adversary queries the secret keys of the attribute from the challenger. The challenger records the attributes provided by the adversary in the list and stores it with the corresponding adversary address in the attribute list.

Challenge phase: here the challenger picks out two identical-length messages (B0, B1) $\in$ Zp. Then the challenger selects $\beta \in \{0, 1\}$ and forwards B β under matrix (T^*, m) to challenger δ.

Stage 2: the adversary inquires about the secret key with same input queries in Stage 1. Guess: the guess that the adversary creates is equal to ½, the probability of guessing $\beta 0$ of β. The game is defined as [Fs $[\beta 0 = \beta] - 1/2$]. Thus our model is secure against selective ciphertext attack.

8.3. Data Security. The adversary is unable to obtain to decrypt the ciphertext as its attribute must satisfy the access policy defined in the matrix structure corresponding to a row of Tm. For unauthorized set of rows L, there exists a vector $\omega.(1, 0, \ldots..0) = -1$ and $\omega.M_i = 0$ for $i \in\in L$, where $L = \{\mathrm{i}: m_\mathrm{i} \in S\}$.s and ω is the polynomial time in matrix M. The adversary cannot calculate the first element of vector ω. Thus, the proposed scheme ensures the data security.

8.4. Forward Security. The attribute authority revokes the attribute of the user with the users address and RSA key pair. The user/address of the revoked attribute cannot decrypt the data again as AA has deleted/blocked the address of the corresponding attribute from the attribute list. Hence the proposed method ensures forward security.

8.5. Collision Resistance. The proposed method ensures resistance to collision attack. The hierarchical access structure policy has been defined in the system. There is a chance of colliding with same access policy generated by multiple users. In this system, a unique address is generated for each set of attributes defined by different users. Hence address of $\mathrm{User_A} \neq$ address of $\mathrm{User_B}$. This uniqueness provides collision-free system.

8.6. Data Integrity. The owner of the data computes ciphertext with symmetric encryption key algorithm E to encode the plaintext message B using symmetric key c_r and calculates the ciphertext data $CT_d = E_{c_r}(B)$. Then the hash value on the ciphertext data $H_{CT} = H(CT_d)G$ is calculated. This ensures data integrity. The final cryptic message uploaded to the cloud is $CT = \{(M, m), C_0, \{C_{1,x}, C_{2,x}, C_{3,x},\}, CT_d, H_{CT}\}$. During decryption, the hash value is used to ensure data integrity at the data requester end.

8.7. Unauthorized Communication. In each level, the data is decrypted using the user address and an RSA key pair. For each set of attributes defined by distinct users, a unique address is generated in this system. As a result, user A's address is different from user B's address and the unauthorized users cannot be entirely in the system. This prevents unauthorized hierarchical communication between nodes.

9. Performance Evaluations

This section briefs the performance evaluation in terms of used system properties, communication overhead, and computation overhead with the previous works and the proposed work.

The subsequent section describes Table 1 that provides the comparative study of the proposed approach with other research works carried out.

From Table 1, it is found that all the works carried out rely on ECC scheme rather than bilinear pairing and [24, 44] use LSSS based access structure where [21, 42, 43] use AND gate access structure whose performance is lesser than the previous LSSS approach. The proposed work differs from all the above by adopting hierarchical access structure with LSSS. Each attribute set corresponds to a row of a matrix in LSSS structure. Thus, our scheme supports multiauthority and multiple data access method.

Table 2 describes the computation cost encountered in current research study and compared study. The entire scheme employs common ECC with 160 bit by $|G|$. It seems that our scheme considers a single row in matrix structure for performing cryptographic operation which is more efficient than the compared schemes.

It is observed that, from [21, 43], the computation overhead depends on the difference between the number of attributes defined in the access policy and the total attributes in the system. Schemes in [24, 44] depend on different attribute set to perform cryptographic process. The scheme in [42] uses KP-ABE and our scheme uses CP-ABE and has

Table 1: Performance comparison of the proposed study with other research studies.

Research study	Pairing free	Access structure	Multiauthority	Decryption outsourcing
[21]	**Yes**	**AND gate**	**No**	**No**
[24]	**Yes**	**LSSS**	**No**	**No**
[42]	**Yes**	**AND gate**	**No**	**No**
[43]	**Yes**	**AND gate**	**No**	**Yes**
[44]	**Yes**	**LSSS**	**Yes**	**Yes**
Proposed work	**Yes**	**Hierarchical LSSS**	**Yes**	**Yes**

Table 2: Computation cost of the compared algorithms.

Scheme	Encryption	Predecryption	Local decryption
[21]	$(Nt - \|\Lambda\| + 2)g$	—	$(Nt - \|\Lambda\| + 3)g$
[24]	$(3\,Ns + 1)g$	—	$(Dt + 1)g$
[42]	$(Ns + 1)g$	—	$(Dt + 1)g$
[43]	$(Nt - \|\Lambda\| + 1)g$	—	$(Nt - \|\Lambda\| + 2)g$
[44]	$(4\,Ns + 1)g$	$(Dt + 1)g$	$(1)g$
Proposed work	$(Ns + 1)g$	$(Dt + 1)g$	$(1)g$

Ns: total number of rows in the matrix Λ, Nt: system attributes, Dt: number of rows in access matrix Λ, Nt: total number of attributes in the system, Dt: number of attributes fulfilling the access policy, |Λ|: access policy attributes, and g: ECC scalar factor for multiplication.

Table 3: Communication overhead of the compared algorithms.

Scheme	Private key (bits)	Public key (bits)	Ciphertext (bits)
[21]	$2^{*}\|g\|$	$(3Nt + 1)\|g\|$	$(Nt - \|\Lambda\| + 3)\|g\|$
[24]	$(\|\Lambda\|)\|g\|$	$(2Nt + 2)\|g\|$	$(2\,Ns + 1)\|g\|$
[42]	$(Ns + 1)\|g\|$	$(2Nt + 2)\|g\|$	$(2\,Ns + 2)\|g\|$
[43]	$1^{*}\|g\|$	$(Nt + 1)\|g\|$	$(Nt - \|\Lambda\| + 2)\|g\|$
[44]	$(\|\Lambda\|)\|g\|$	$(2Nt + 2)\|g\|$	$(3\,Ns + 1)\|g\|$
Proposed work	$(\|\Lambda\|)\|g\|$	$(Nt+\ 2)\|g\|$	$(Ns + 1)\|g\|$

Ns: total number of rows in the matrix Λ, Nt: system attributes, Dt: number of rows in access matrix Λ, Nt: total number of attributes in the system, Dt: number of attributes fulfilling the access policy, |Λ|: access policy attributes, and g: ECC scalar factor for multiplication.

efficient encryption process as the encryption process depends on set of attributes alone. But there is an overhead in decryption process in the scheme in [42] as the end node has to do total decryption process. But our scheme offloads the predecryption process to AA and final decryption at the end nodes. The proposed work excels other compared works in computation and communication cost.

Table 3 discusses the communication cost observed in our work and other compared works. Compared with other works, our scheme and the scheme in [44] have increased communication overhead as predecryption process is carried at the AA in our scheme and at edge nodes in the scheme in [44]. Also revocation process is carried at AA in our scheme without affecting other components in the system. Hence the communication overhead at the end data requester is minimum compared with others.

To excel our scheme, we implemented our proposed work in Ubuntu platform. It is deployed with Python and charm library to implement the ECC using simple scalar multiplication. The scheme is implemented with 512-bit ECC where 160 bits serves as ECC group order. In Figures 4 and 5, we show that our scheme excels the work done in [24, 44] in executing cryptographic operations. Our scheme excels [24, 44] in executing encryption method and decryption outsourcing is performed in the proposed work and [44]; hence it has same decryption time.

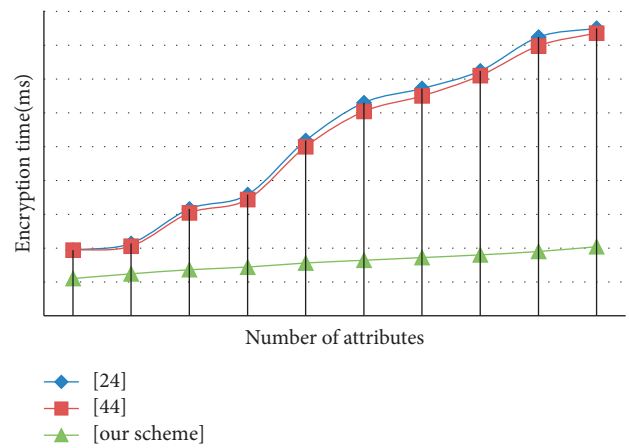

Figure 4: Comparison of encryption time (ms).

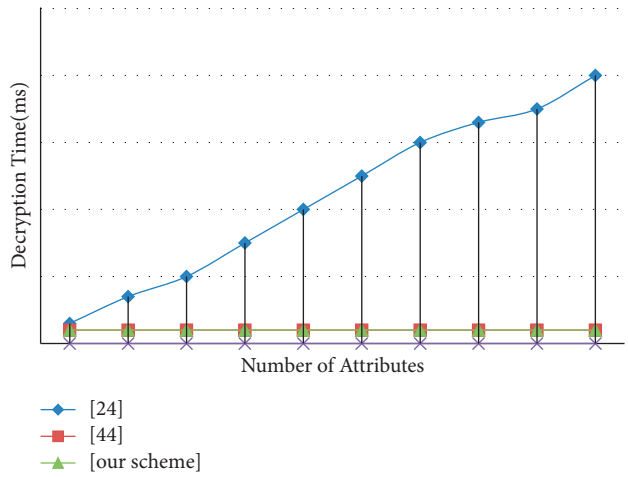

Figure 5: Comparison of decryption time (ms).

10. Conclusion

The proposed work uses CP-ABE using ECC in blockchain network with hierarchical access structure. The scheme considers multiple authorities and multiple data access by defining the attribute set. The attribute set is represented as row in LSSS matrix structure. For each attribute set, a unique address is generated along with RSA key pair. This pair is helpful in revoking the attribute, thereby providing security from unauthorized users. Further, the security mechanism of the proposed work is defined under d-DDH assumption. From the experimental analysis, it is found that our scheme shows better outcome than the compared work. The constraint of the current work affects the ciphertext length with increase in number of attributes. Hence, in future work an efficient CP-ABE will focus on alleviating this problem.

Data Availability

The data shall be made available on request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. F. Ransome, *Cloud Computing: Implementation, Management, and Security*, CRC Press, Boca Raton, Florida, 2017.

[2] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Transactions on Information and System Security*, vol. 9, no. 1, pp. 1–30, 2006.

[3] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, "March). Plutus: scalable secure file sharing on untrusted storage," *Fast Company*, vol. 3, pp. 29–42, 2003.

[4] S. D. C. Di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Over-encryption: management of access control evolution on outsourced data," in *Proceedings of the 33rd International Conference on Very Large Data Bases*, pp. 123–134, Vienna, Austria, 2007, September.

[5] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP'07)*, pp. 321–334, IEEE, Berkeley, CA, USA, 2007, May.

[6] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption," in *Advances in Cryptology - EUROCRYPT 2010*, pp. 62–91, Springer, Berlin, Heidelberg, 2010.

[7] H. Deng, Q. Wu, B. Qin et al., "Ciphertext-policy hierarchical attribute-based encryption with short ciphertexts," *Information Sciences*, vol. 275, pp. 370–384, 2014.

[8] X. Yan, H. Ni, Y. Liu, and D. Han, "Privacy-preserving multi-authority attribute-based encryption with dynamic policy updating in PHR," *Computer Science and Information Systems*, vol. 16, no. 3, pp. 831–847, 2019.

[9] L. Aceto, I. Damgaard, L. A. Goldberg, M. M. Halldorsson, A. Ingolfsdottir, and I. Walukiewicz, *Automata, languages and programming: 35th international colloquium, ICALP 2008 Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II*, Vol. 5126, Springer, , Berlin, Heidelberg, 2008.

[10] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of abe ciphertexts," in *Proceedings of the USENIX security symposium*, vol. 2011, no. 3, San Diego, CA, USA, 2011, August.

[11] W.-M. Li, X.-L. Li, Q.-Y. Wen, S. Zhang, and H. Zhang, "Flexible CP-ABE based access control on encrypted data for mobile users in hybrid cloud system," *Journal of Computer Science and Technology*, vol. 32, no. 5, pp. 974–990, 2017.

[12] P. Zhang, Z. Chen, J. K. Liu, K. Liang, and H. Liu, "An efficient access control scheme with outsourcing capability and attribute update for fog computing," *Future Generation Computer Systems*, vol. 78, pp. 753–762, 2018.

[13] C. Zuo, J. Shao, G. Wei, M. Xie, and M. Ji, "CCA-secure ABE with outsourced decryption for fog computing," *Future Generation Computer Systems*, vol. 78, pp. 730–738, 2018.

[14] H. Zhong, Y. Zhou, Q. Zhang, Y. Xu, and J. Cui, "An efficient and outsourcing-supported attribute-based access control scheme for edge-enabled smart healthcare," *Future Generation Computer Systems*, vol. 115, pp. 486–496, 2021.

[15] C. Chen, Z. Zhang, and D. Feng, "Efficient ciphertext policy attribute-based encryption with constant-size ciphertext and constant computation-cost," in *International Conference on Provable Security*, pp. 84–101, Springer, Berlin, Heidelberg, 2011.

[16] Z. Zhou, D. Huang, and Z. Wang, "Efficient privacy-preserving ciphertext-policy attribute based-encryption and broadcast encryption," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 126–138, 2013.

[17] F. Fuchun Guo, Y. Yi Mu, W. Susilo, D. S. Wong, and V. Varadharajan, "CP-ABE with constant-size keys for lightweight devices," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 5, pp. 763–771, 2014.

[18] P. S. L. M. Barreto, H. Y. Kim, B. Lynn, and M. Scott, "Efficient algorithms for pairing-based cryptosystems," in *Advances in Cryptology - CRYPTO 2002*, pp. 354–369, Springer, Berlin, Heidelberg, 2002.

[19] K. Lauter, "The advantages of elliptic curve cryptography for wireless security," *IEEE Wireless communications*, vol. 11, no. 1, pp. 62–67, 2004.

[20] M. Zheng, Y. Xiang, and H. Zhou, "A strong provably secure IBE scheme without bilinear map," *Journal of Computer and System Sciences*, vol. 81, no. 1, pp. 125–131, 2015.

[21] 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, 2016.

[22] V. Odelu, A. K. Das, M. Khurram Khan, K.-K. R. Choo, and M. Jo, "Expressive CP-ABE scheme for mobile devices in IoT satisfying constant-size keys and ciphertexts," *IEEE Access*, vol. 5, pp. 3273–3283, 2017.

[23] K. Sowjanya, M. Dasgupta, S. Ray, and M. S. Obaidat, "An efficient elliptic curve cryptography-based without pairing KPABE for Internet of Things," *IEEE Systems Journal*, vol. 14, no. 2, pp. 2154–2163, 2019.

[24] 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, 2018.

[25] N. Raj and A. R. Pais, "CP-ABE scheme satisfying constant-size keys based on ECC," *ICETE*, vol. 2, pp. 535–540, 2020.

[26] Y. Tian, T. Shao, and Z. Li, "An efficient scheme of cloud data assured deletion," *Mobile Networks and Applications*, vol. 26, pp. 1–12, 2020.

[27] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pp. 456–465, Alexandria, Virginia, USA, 2007, October.

[28] B. Waters, "Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization," in *Public Key Cryptography - PKC 2011*, pp. 53–70, Springer, Berlin, Heidelberg, 2011.

[29] R. Bharti, A. Khamparia, M. Shabaz, G. Dhiman, S. Pande, and P. Singh, "Prediction of heart disease using a combination of machine learning and deep learning," in *Computational Intelligence and Neuroscience*, A. A. Abd El-Latif, Ed., vol. 2021pp. 1–11, 2021.

[30] H. He, J. Zhang, J. Gu, Y. Hu, and F. Xu, "A fine-grained and lightweight data access control scheme for WSN-integrated cloud computing," *Cluster Computing*, vol. 20, no. 2, pp. 1457–1472, 2017.

[31] P. Ratta, A. Kaur, S. Sharma, M. Shabaz, and G. Dhiman, "Application of blockchain and Internet of Things in healthcare and medical sector: applications, challenges, and future perspectives," in *Journal of Food Quality*, R. Khan, Ed., vol. 2021pp. 1–20, 2021.

[32] J. Li, W. Yao, Y. Zhang, H. Qian, and J. Han, "Flexible and fine-grained attribute-based data storage in cloud computing," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 785–796, 2016.

[33] J. Li, Y. Zhang, X. Chen, and Y. Xiang, "Secure attribute-based data sharing for resource-limited users in cloud computing," *Computers & Security*, vol. 72, pp. 1–12, 2018.

[34] W. Teng, G. Yang, Y. Xiang, T. Zhang, and D. Wang, "Attribute-based access control with constant-size ciphertext in cloud computing," *IEEE Transactions on Cloud Computing*, vol. 5, no. 4, pp. 617–627, 2015.

[35] F. Ajaz, M. Naseem, S. Sharma, M. Shabaz, and G. Dhiman, "COVID-19: challenges and its technological solutions using IoT," in *Current Medical Imaging Formerly: Current Medical Imaging Reviews*vol. 17, , 2021.

[36] T. Naruse, M. Mohri, and Y. Shiraishi, "Provably secure attribute-based encryption with attribute revocation and grant function using proxy re-encryption and attribute key for updating," *Human-centric Computing and Information Sciences*, vol. 5, no. 1, pp. 1–13, 2015.

[37] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, "User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1767–1777, 2017.

[38] L. Xue, Y. Yu, Y. Li, M. H. Au, X. Du, and B. Yang, "Efficient attribute-based encryption with attribute revocation for assured data deletion," *Information Sciences*, vol. 479, pp. 640–650, 2019.

[39] Y. Wang, F. Li, J. Xiong, B. Niu, and F. Shan, "Achieving lightweight and secure access control in multi-authority cloud," in *IEEE Trustcom/BigDataSE/ISPA*vol. 1, , pp. 459–466, IEEE, 2015.

[40] M. Chase, "Multi-authority attribute based encryption," in *Theory of Cryptography Conference*, pp. 515–534, Springer, Berlin, Heidelberg, 2007.

[41] S. Wang, J. Zhou, J. K. Liu, J. Yu, J. Chen, and W. Xie, "An efficient file hierarchy attribute-based encryption scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1265–1277, 2016.

[42] 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, 2015.

[43] A. K. Junejo and N. Komninos, "A lightweight Attribute-based security scheme for fog-enabled cyber physical systems," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 2145829, 18 pages, 2020.

[44] 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, p. 1176, 2021.

[45] W. Wang, H. Xu, M. Alazab, T. R. Gadekallu, Z. Han, and C. Su, "Blockchain-based reliable and efficient certificateless signature for IIoT devices," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, 2021.

[46] W. Wang, C. Qiu, Z. Yin et al., "Blockchain and PUF-based lightweight authentication protocol for wireless medical sensor networks," *IEEE Internet of Things Journal*, vol. 14, no. 8, 2021.

[47] H. Xiong, C. Jin, M. Alazab et al., "On the design of blockchain-based ECDSA with fault-tolerant batch verication protocol for blockchain-enabled IoMT," *IEEE Journal of Biomedical and Health Informatics*, 2021.

[48] L. Zhang, Y. Zou, W. Wang, Z. Jin, Y. Su, and H. Chen, "Resource allocation and trust computing for blockchain-enabled edge computing system," *Computers & Security*, vol. 105, Article ID 102249, 2021.

[49] D. Liu, Y. Zhang, W. Wang, K. Dev, and S. A. Khowaja, "Flexible data integrity checking with original data recovery in IoT-enabled maritime transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2021.