Multiauthority CP-ABE-based Access Control Model for IoT-enabled Healthcare Infrastructure

Article *in* IEEE Transactions on Industrial Informatics · January 2022

DOI: 10.1109/TII.2022.3167842

2 authors:

Sangjukta Das
National Institute of Technology Patna
4 PUBLICATIONS **92** CITATIONS

SEE PROFILE

Suyel Namasudra
National Institute of Technology, Agartala
78 PUBLICATIONS **2,035** CITATIONS

SEE PROFILE

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3167842, IEEE Transactions on Industrial Informatics

Multi-Authority CP-ABE-Based Access Control Model for IoT-Enabled Healthcare Infrastructure

Sangjukta Das and Suyel Namasudra, *Member, IEEE*

***Abstract*—The exponential growth of the Internet of Things (IoT) technologies requires high data security. Here, data security is very critical as all IoT devices transfer data over the internet. The fine-grained access control provided by the Ciphertext Policy Attribute-Based Encryption (CP-ABE) technique can be considered as a potential solution to this issue. However, CP-ABE uses bilinear pairing operation for its internal working, which is expensive for any resource constraint device. An Elliptic Curve Cryptography (ECC) based CP-ABE scheme can be well suited for resource constraint IoT framework because ECC takes less computational time. This paper proposes a novel CP-ABE technique based on ECC to achieve fine-grained access control over data or resources. The proposed technique includes multiple attribute authorities to manage attributes and key generation, which can reduce the work overhead of having a single authority in traditional CP-ABE systems. In addition, the proposed scheme outsources the decryption process to a user assistant entity to reduce the decryption overhead of the end-users. To prove the efficiency of the proposed scheme, both formal security analysis and performance comparisons are presented in this paper. The result and findings prove the effectiveness of the proposed scheme over some well-known schemes.**

***Index Terms*—Attribute Based Encryption, Elliptic Curve Cryptography, Re-Encryption, Cloud Service Provider.**

I. Introduction

THE IoT describes an intelligent network comprising enormous heterogeneous inter-connected devices, which communicate with each other over the internet. The "things" in IoT refer to the smart devices embedded with sensors, software, and other technologies. IoT has a significant impact in almost every industry, including the healthcare industry with its enhanced automation, optimization, and analytical features. In this advanced technology, data collection, management, and processing are much easier with the help of sensors or things embedded in smart devices. However, most IoT devices have the low processing power, limited battery life, low communication capabilities, and restricted storage capacity, i.e., devices are resource-constrained [1]. Therefore, storing data in a cloud environment is generally practiced by many organizations. However, the security of data in a cloud environment completely depends on the security strategy used by the Cloud Service Provider (CSP) [2]. A malicious cloud environment can disclose patients' sensitive data to attackers to gain economic or commercial benefits [3]. During data transmission, attackers can perform attacks, such as eavesdropping, impersonation, collusion attack, etc. [4]. So, data encryption, as well as access control over data before transmission, is suggested by many researchers to prevent data from unwanted cyber-attacks. The security and privacy of data can be enhanced by controlling the access to the data or resource [5].

In a healthcare infrastructure, there can be various types of information, such as sleep time, BP rate, pulse rate, heart rate, temperature, oxygen saturation, and many more. Unauthorized access to this information can have devastating consequences for the patient's health. Therefore, while building a healthcare infrastructure, it is important to restrict unauthorized user's access [6]. For example, a particular hospital's authorized doctor or healthcare professional can only access a patient's data during business hours. This condition cannot be satisfied by using typical public-key encryption with the one-to-one access mode. Instead, the above-mentioned issue can be solved by using an encryption system with one-to-many access mode and fine-grained access control. Many strategies for controlling data access have been proposed in the literature. Identity-Based Encryption (IBE) techniques can improve the security of healthcare data in a cloud-assisted IoT-based healthcare system [7]. Here, data is encrypted using the intended recipient's public identity (e.g. email ID). Attribute-Based Encryption (ABE) is an extension of IBE that allows encryption and decryption of a message based on attribute set and access structures. ABE-based schemes can support access control along with a data encryption scheme for multiple users at the same time [8]. However, a role-based access control policy needs to be defined to gain fine-grained access control over healthcare data. According to [9, 10], CP-ABE-based schemes can define the user's role using a set of attributes. In addition, fine-grained access control over healthcare data can also be achieved by restricting access based on the roles of individual users.

Zhang et al. [11] have presented a CP-ABE scheme for personal health record systems, which makes the access policy more expressive by using the Linear Secret Sharing Scheme (LSSS) with multiple values. In [10, 11], one single authority distributes private-public key pairs to the data customers. As a result, these systems face the key-escrow problem, which mainly occurs if a system has a fully trusted key generation or

S. Das and S. Namasudra are with the Department of Computer Science and Engineering, National Institute of Technology Patna, Bihar, India. Email: {sangjukta24, suyelnamasudra}@gmail.com

managing authority. Then, this authority can decrypt the encrypted message by using the keys generated by itself. Chase et al. [12] have presented the first Multi-Authority ABE (MA-ABE) scheme in which multiple independent attribute authorities maintain their own set of attributes and generate secret keys for the user rather than relying on a single authority. Yan et al. [13] have proposed another MA-ABE scheme, where one partially hidden access policy is used to prevent the user's privacy and attribute values from being disclosed to third parties. Additionally, the access structure and policy-updating algorithms of the LSSS are designed to enable all sorts of policy updates. Zhang et al. [14] have proposed another multi-authority ABE scheme based on bilinear pairing for sharing healthcare data securely. Here, encrypted data are decrypted partially on an outsourced entity to reduce the decryption overhead of the data users. The CP-ABE schemes proposed in [15-18] use complex bilinear pairing operations, which make these schemes difficult for resource constraint data users to efficiently decrypt messages. Moreover, these schemes do not solve the key-escrow problem. These problems restrict the implementation of the CP-ABE scheme in IoT-based healthcare systems. Ding et al. [19] have proposed a pairing-free CP-ABE scheme using ECC for IoT-enabled systems. This scheme replaces computationally expensive bilinear pairing by using elliptic curve-based calculations. Yao et al. [20] have proposed a Lightweight no-pairing ECC-based ABE (LEABE) scheme for IoT infrastructure. Although LEABE has less communication and computational overhead, it has poor flexibility and scalability. This scheme is further extended by Qin et al. [21] to propose an ECC-Based Access Control (EBAC) scheme for vehicular networks. EBAC has achieved mutual authentication between authority and user by including an implicit certificate and lightweight decryption by outsourcing the decryption process. However, both LEABE and EBAC are based on Key Policy-based ABE (KP-ABE), which is not ideal for IoT systems. Cheng et al. [22] have proposed an Efficient ECC-based CP-ABE (EECPABE) scheme to support lightweight decryption, which is ideal for IoT systems with limited resources. Apart from the schemes discussed above, there are many existing schemes [23-25] in the literature, which improve the security and performance of IoT systems. However, most of these existing schemes do not guarantee data confidentiality, security, access control, or users' privacy, which motivates to develop a scheme that can provide security, as well as fine-grained access control to healthcare data.

In this paper, a novel scheme has been proposed for IoT-based healthcare infrastructure, namely Lightweight Access Control Model using Multiple Attribute Authority (LACMMAA), to solve the problems of the existing schemes. The proposed scheme is based on CP-ABE and ECC, and here, multiple authorities are responsible for generating keys related to the user's attribute set. After collecting data from IoT devices, the Data Owner (DO) encrypts the data and defines an access policy for the authorized set of users. Then, the DO uploads this data to the cloud service provider. The CSP re-encrypts this encrypted data and stores it on the cloud storage. Whenever a Data User (DU) requests for data, the CSP sends the ciphertext and the key to the DU. Here, each DU gets registered with the Central Authority (CA) before sending a data request to the CSP. To reduce decryption overhead at the end-user, the proposed scheme outsources the decryption process to a Data User Assistant (DUA), which partially decrypts the ciphertext before executing the complete decryption process at the DU.

The key contributions of this work are concise as follows.

1) A novel scheme has been proposed in this paper based on elliptic curve cryptography and CP-ABE for fine-grained access control in IoT-based healthcare systems.
2) LACMMAA reduces the key generation overhead by distributing the work into multiple authorities, and also, overcomes the key-escrow problem that improves security.
3) Here, an entity is used to outsource the decryption process that reduces the decryption overhead of the end-users.
4) Results and discussion, as well as security analysis of the proposed scheme prove its efficiency.

The rest of the parts of this paper are organized in the following manner. A few preliminary studies are discussed in section II. Section III and IV represent the overview and the construction of the proposed scheme, respectively. The security analysis is discussed in section V. Next, section VI evaluates the performance of the proposed scheme. Finally, section VII concludes the whole paper with some future goals.

II. Preliminary Studies

A. Elliptic Curve Cryptography

Let F_q be a prime finite field of integer modulo q, and E be an elliptic curve over F_q. The curve E is shown by Eq. (1).

$$E: y^2 \ (m \quad q) = x^3 + a \ + b \ (m \quad q) \tag{1}$$

where $a, b, x, y \in E$ and $4a^3 + 27b^2 \neq 0$. F_q is also referred to as $G \ (q)$. G is a cyclic group of order r, contains all the points on E. The generator G, which is a point in G , can generate any other point in its cyclic subgroup by multiplying G by an integer i such that $i < r$. The cofactor h is the number of subgroups holding all the curve points such that $h = E(F_q)/r$. Here, $(q, a, b, G, r,$ and $h)$ are the domain parameters of E. All the notations used in this paper are described in Table I.

Suppose, two users U_a and U_b want to communicate with each other. They choose private keys as p_a and p_b, respectively, and calculate corresponding public keys q_a and q_b as follows:

$$q_a = p_a * G \text{ and } q_b = p_b * G$$

To send a plaintext message P to U_b, U_a uses U_b's public key q_b for encrypting P as shown in Eq. (2).

$$C \ = [k.G, (P \ + k.q_b)] \tag{2}$$

where $k \in z_r$ is an integer chosen randomly by U_a and $z_r = \{0, 1, \dots, r-1\}$. On receiving the C , the receiver, i.e. U_b decrypts the C using the private key p_b as shown in Eq. (3).

$$P \ = (P \ + k.q_b) - k.G.p_b \tag{3}$$

B. Decisional Diffie-Hellman (DDH) Problem

Considering a challenger chooses a cyclic group G of order r and a generator point G. Now, it selects $a, b \in z_r$ randomly. If the challenger sends a tuple $(a \ , b \ , G)$ to an adversary, as per the DDH assumption, it is difficult for the adversary to correctly guess $a \quad \in G$ from a random element $R \in G$ [19].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3167842, IEEE Transactions on Industrial Informatics

Definition 1: The DDH assumption holds if a hacker has a negligible advantage to solve the DDH problem in a polynomial time.

C. Access Structure and Attribute

If there is a set of n parties, then, an access structure $\mathbb{A}' \subseteq 2^{\{F_1,F_2,\dots,F_n\}}$ is a collection of non-empty sets of $\{F_1, F_2, \dots, F_n\}$ [21]. This can be transformed into a boolean function, which again can be represented by an access tree, where attributes are represented as leaf nodes and the logical operators (AND or OR) are represented as root or intermediate nodes.

Considering a set of n attributes $A_1, A_2, \dots, A_n$ that make up the attribute universe, $U = \{A_1, A_2, \dots, A_n\}$. A is a user's attribute set, represented by an n-bit string $a_1, a_2, \dots, a_n$, specified as $a_i = 0$, if $A_i \notin A$, and $a_i = 1$, if $A_i \in A$. Γ specifies an access policy with attributes in U, which can be represented by n-bit string $p_1, p_2, \dots, p_n$ defined as follows: $p_t = 0$, if $A_t \notin \Gamma$, and $p_t = 1$, if $A_t \in \Gamma$.

Considering an example, where $A = \{A_1, A_3, A_5\}$ is the user's attribute set and $n = 5$. Then, A can be represented as a 5-bit string 10101. Consider another example, where $n = 5$ and $\Gamma = 01011$. This specifies that the attribute set $\{A_2, A_4, A_5\}$ is required to satisfy the access policy Γ.

TABLE I
DESCRIPTION OF NOTATIONS

Notation	Description	Notation	Description
α	Attribute set	$G_1(q)$	Finite field of order q
$(\mathbb{A}, \rho)$	Access policy	$\mathbb{A}_x$	x^{th} row of matrix $\mathbb{A}$
G	General identity	D_D	Private key of the DO
c_s	CSP's secret key	Q_D	Public key of the DO
C_{sP}	CA's private key	D_u	Private key of the DU
C_{sP}	CA's public key	Q_u	Public key of the DU
PK	Public key	MSK	Master secret key
r_u	DU's reconstruction parameter	γ, β, K_c	Random numbers chosen by the CA

D. Linear Secret Sharing Scheme

A secret sharing scheme is a cryptographic primitive to share a secret to n parties in such a way that each party has a share of that secret. The original secret cannot be revealed unless the shares of all parties are combined. A secret sharing is a linear scheme, if the reconstruction of the original secret is done by linear mapping from the shares. An LSSS on z_r based on n parties $\{F_1, F_2, \dots, F_n\}$ should satisfy the following properties:

1) The secret share of each F_i can form a vector over z_r.
2) $\mathbb{A}$ is a sharing matrix of size $(l \times m)$, where each row $x \in [1, 2, \dots, l]$ represents a party in the set F. A function $\rho(x)$ indicates to the attribute in $\mathbb{A}_x$. To share a secret message $s \in z_r$, a random vector $v = (s, v_1, \dots, v_m)$ is considered. Then, the vector $(\mathbb{A}_x . v) \rightarrow \lambda_x$, where $x \in [1, 2, \dots, l]$, divides s into l shares according to the LSSS.
3) For any authorized set of attributes $\alpha \in \mathbb{A}'$, one constant set $c_x \in z_r$ can be generated in a polynomial time. By using c_x and l number of shares λ_x, the secret s can be reconstructed by calculating $\sum_{x \in [1,2,\dots,l]} c_x . \lambda_x = s$.

III. OVERVIEW OF THE PROPOSED SCHEME

The proposed scheme's system model, system definitions, and design goals are represented in the following subsections.

A. System Model

There are mainly eight entities in the proposed scheme, namely Data Owner (DO), Central Authority (CA), Cloud Service Provider (CSP), Attribute Authority (AA), Data User Assistant (DUA), Data User (DU), IoT Devices, and Gateways.

1) *Data Owner:* The DO stores data received from gateways in its local storage and defines an access policy. Then, it encrypts the data using the defined access policy before outsourcing the data to the cloud storage.
2) *Central Authority:* The CA is in charge of setting the system initially and generating the public parameters. The CA registers each user and maintains a list containing user' details, which are used for verifying the user's authenticity.
3) *Cloud Service Provider:* The CSP is a semi-trusted entity. It offers data-sharing services to the entire system.
4) *Data User Assistant:* The DUA is responsible for partial decryption of the ciphertext and returns the partially decrypted data to the DU.
5) *Data User:* A user may be a patient, physician, or any person, who has the authorization to access any data. The DU can decrypt the data, if and only if it has a sufficient number of attributes to satisfy the access policy.
6) *Attribute Authority:* Each AA is responsible for generating the private and public key pair for an attribute set that belongs to its domain. The AA also generates the user secret key related to the user's attribute.
7) *IoT Device:* IoT devices are connected with the gateways through a wired or wireless medium. These devices collect data from patients and send the collected data securely to the gateway to which it is connected.
8) *Gateways:* A gateway receives collected data from IoT devices. These data are then aggregated at the gateway and sent to the data owner through a secured channel.

B. System Definitions

The working of the proposed scheme is represented in this subsection, which consists of seven algorithms:

$Sys\_Setup(q \rightarrow P)$: The CA executes the *Sytem_Setup* algorithm for setting the system. It selects a large prime number q as security parameter and outputs all Public Parameters (P).

$AA\_Setup\{P \rightarrow (P, M)\}$: The AA performs this algorithm by taking P as input and outputs (P, M) for its attribute set.

Registration $\{(R_u, I_u) \rightarrow G\}$: On receiving a registration request from the DU, the CA generates G for each DU by using (R_u, I_u) as inputs.

$Key\_Gen\{(M, \alpha) \rightarrow U_S\}$: The AA and DU generate the user's secret key U_S for each user by using this *Key_Gen* algorithm, which takes α and M as inputs.

$Encrypt\{(P, c_s, P, (\mathbb{A}, \rho)) \rightarrow C\}$: The DO encrypts message P under $(\mathbb{A}, \rho)$ by taking P, P, c_s, and $(\mathbb{A}, \rho)$ as inputs and outputs the ciphertext C.

$Re\_Encrypt\{(C, c_s) \rightarrow C_c\}$: The CSP re-encrypts the encrypted data C as C_c by using c_s before storing it.

$Dec\{(P, C_c, U_S, c_s, (\mathbb{A}, \rho)) \rightarrow P\}$: The DU and DUA perform this algorithm by using P, C_c, U_S, c_s, and $(\mathbb{A}, \rho)$ to decrypt the C_c for generating P.

C. Design Goals

There are mainly three design goals of the proposed scheme as discussed below:

1) *Security:* Nowadays, there are a huge number of attackers over the internet, who attempt to hack confidential healthcare data. Thus, the security of healthcare data is one of the major requirements.
2) *Fine-Grained Access Control:* Access to any healthcare data in a healthcare system must be controlled to prevent any unauthorized and illegal user from accessing the data. Fine-grained access control can be enforced through access policy to allow or deny access to certain healthcare data.
3) *Lightweight Technique:* *IoT devices are always resource-constraint. Therefore, it is necessary to design a lightweight IoT healthcare infrastructure, which requires low computational cost and less power consumption.*

IV. Construction

In the proposed scheme, the central authority initializes the system and registers all the entities based on their requests.

A. System Setup

The *System_Setup* algorithm takes a sufficiently large prime number q to generate P as $(q, G\ (q), E, a, b, G, r, h)$ for the system. Fig. 1 depicts the proposed scheme's entire workflow.

B. Authority Setup

Each of the multiple AAs takes P as input to perform the *Authority_Setup* algorithm, where it selects integers $y_i, k_i \in z_r$ randomly as MSK and calculates the PK as $\{y_i.G, k_i.G\}$ for the attribute set that belongs to its domain.

C. Registration

When a user makes a registration request to the CA to join the IoT-based healthcare system, the CA first checks users' details. Then, the CA along with the DU generates Login Credential (LC) and private-public key pair of the DU. A user is permitted to login into the system only after a successful registration phase. The CA maintains a list containing the users' IDs and corresponding details, which are used for verifying individual identity and mutual authentication. After receiving GID, the DU can verify whether the message is received from the CA or not. The steps involved in this process are as follows:

Step 1: At first, each DU chooses a number K_u randomly from z_r such that $K_u \in z_r$ and computes $R_u = K_u.G$. Then, the DU sends R_u and its real identity (ID) to the CA to get registered.
Step 2: On getting the details from the DU, the CA selects a random number K_c from z_r to calculate $P_c = R_u + K_c.G$.
Step 3: Then, the CA again randomly chooses $\in$ and β from z_r and computes $G = \{(I + \in.\beta.G), \in.G\}$.
Step 4: In the fourth step, the CA generates $C_U = E\ (P_c, G)$ and calculates hash value, $h_1 = ha\ h(C_U)$ and $r_u = h_1.K_c + C_P$.
Step 5: Then, C_U and r_u are sent to the DU.
Step 6: On receiving the details of step 5, the DU computes $h_1 = ha\ h(C_U)$, $D_u = h_1.K_u + r_u$, and $Q_u = D_u.G$.
Step 7: In this step, the DU finally obtains P_{ca} and G from C_U by calculating $D\ (C_U)$.

Here, the DU can verify that the received message $\{C_U, r_u\}$ is authentic or not by calculating $Q'_u = h_1.P_c + C_P$ and by checking $Q_u == Q'_u$. If the received message $\{C_U, r_u\}$ is not authentic, the user again sends a registration request to the CA.

D. Key Generation

Here, the A_i calculates the partial user secret key U_S for the DU and maintains a list for it. The A_i sends U_S to the DU. Then, the DU selects a random integer $p \in z_r$ and calculates U'_S, which is used for partial decryption at the DUA. The steps involved in the key-generation process are as follows:

Step 1: At first, the A_i calculates the partial user secret key, $U_{S\ i} = y_i + H(G).k_i$.
Step 2: The A_i sends $U_S = \{U_{S\ i}, i \in a_{j,G}\}$ to the DU.
Step 3: Then, the DU selects a random integer $p \in z_r$.
Step 4: The DU calculates $U'_{sk_i} = y_i + H(G).k_i + p$ and sends U'_S to the DUA

E. Encryption Process

The DO performs the encryption algorithm by selecting a symmetric encryption technique and a symmetric key c_i. Here, c_i is generated as a random string by using a secure pseudo-random number generation process. To encapsulate c_i, the DO chooses $s \in z_r$, and calculates $C_0 = c_i + s.G$. The DO also defines an access policy to restrict user access to the data. The DO again selects two random vectors $v = (s, v_1, \dots, v_m)$ and $u = (0, u_1, \dots, u_m)$, and calculates $\in_x = \mathbb{A}_x.v$ and $\omega_x = \mathbb{A}_x.u$. As shown in step 7, then, the DO calculates $C_0, C_{1,x}$, and $C_{2,x}$ to generate the final ciphertext C_D. To maintain data integrity and prevent the data from being forged, the DO calculates the hash of the C as C_H and sends C_D to the CSP. The steps involved in the encryption process are:

Step 1: At first, the DO selects a symmetric key c_i.
Step 2: Then, the DO encrypts P as $C = e\ (P)_c$.
Step 3: The DO calculates the hash of the C as $C_H = H(C).D_D.G$.
Step 4: The DO then defines an LSSS access policy $(\mathbb{A}, \rho)$.
Step 5: Then, the DO selects a random integer $s \in z_r$ and two random vectors $v = (s, v_1, \dots, v_m)$ and $u = (0, u_1, \dots, u_m)$ to calculate $\in_x$ and ω_x.
Step 6: Here, the DO calculates $\in_x = \mathbb{A}_x.v$ and $\omega_x = \mathbb{A}_x.u$.
Step 7: In the seventh step, the DO calculates $C_0, C_{1,x}$, and $C_{2,x}$.

$$C_0 = c_i + s.G$$
$$C_{1,x} = \in_x.G + y_{\rho(x)}.G,\ C_{2,x} = \omega_x.G + k_{\rho(x)}.G$$

Step 8: Finally, the DO sends $C_D = \{(\mathbb{A}, \rho), C_0, (C_{1,x}, C_{2,x}), C, C_H\}$ to the CSP.

F. Re-Encryption Process

The CSP performs the *Re_Encryption* algorithm by selecting a random integer $c_i \in z_r$ and re-encrypts the C_D as $C_C = E\ (C_D)_c$ before storing C_D on the cloud storage. This process is used to increase the security of the encrypted data on cloud storage. There are the following steps in this phase:

Step 1: The CSP selects a random integer $c_i \in z_r$ as key.
Step 2: The CSP re-encrypts C_D as $C_C = E\ (C_D)_c$.
Step 3: Here, the CSP stores the re-encrypted ciphertext C_C.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3167842, IEEE Transactions on Industrial Informatics

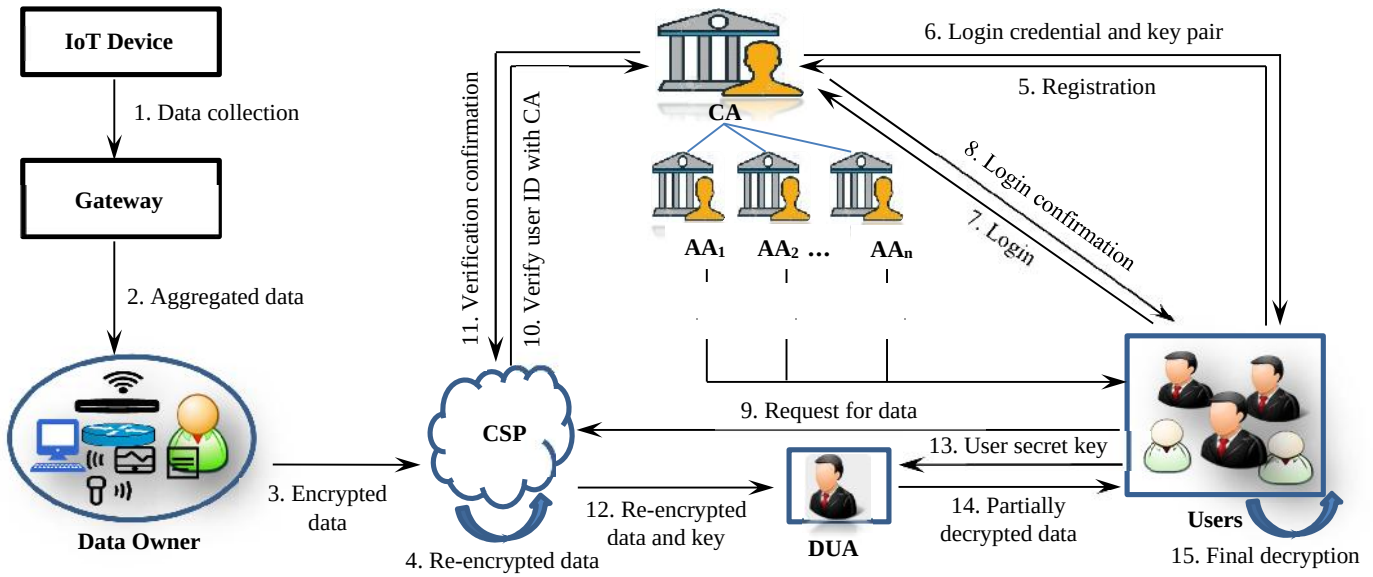

Fig. 1. Proposed scheme's workflow

G. Decryption Process

The decryption process is divided into two phases: 1) partial decryption at the DUA, and 2) final decryption at the DU.

1) *Partial Decryption at the DUA:* After receiving C_{C} and c_i from the CSP, the DUA obtains C_{D} from C_{C}. If u satisfies the access policy $(\mathbb{A}, \rho)$, one constant set $\{c_x \in z_r\}$ can be generated in a polynomial time such that $\sum_{x \in X} c_x . \mathbb{A}_x = (1, 0, \dots, 0) = \varepsilon$. Then, the DUA calculates D_x, N_1, and N_2 as shown in steps 4 and 5, and sends $C'_{D} = \{C_0, C_{H}, N_1, N_2\}$ to the data user.

   Step 1: The DUA obtains C_{D} by calculating $C_{D} = D_{(C_{CP})c_i}$.
   Step 2: In this step, the DUA generates $X = \{x | \rho(x) \in u\}$.
   Step 3: The DUA computes one constant set $\{c_x \in z_r\}$ in a polynomial time such that $\sum_{x \in X} c_x . \mathbb{A}_x = (1, 0, \dots, 0) = \varepsilon$, $\sum_{x \in X} c_x . \gamma_x = s$, and $\sum_{x \in X} c_x . \omega_x = 0$.
   Step 4: Then, the DUA calculates $D_x = C_{1,x} - U_{S_i} . G + H(G_{i}) . C_{2,x} = \gamma_x . G + H(G_{i}) . \omega_x . G - p . G$.
   Step 5: In the fifth step, the DUA again calculates $N_1 = \sum_{x \in X} c_x . D_x = s . G - p . \sum_{x \in X} c_x . G$, and $N_2 = \sum_{x \in X} c_x . G$.
   Step 6: At last, the DUA sends C'_{D} to the DU.

2) *Final Decryption at the DU:* After receiving the partially decrypted ciphertext from the DUA, the DU performs a scalar multiplication in order to generate c_i. The steps involved in the final decryption process are as follows:

   Step 1: The DU calculates $c_i' = C_0 - N_1 - p . N_2 = c_i + s . G - s . G + p \sum_{x \in X} c_x . G - p \sum_{x \in X} c_x . G = c_i$.
   Step 2: In the second step, the DU calculates $H(C_{i}) . Q_D$.
   Step 3: The CT is accepted only if $C_{H} == H(C_{i}) . Q_D$, where $Q_D = D_D . G$. Finally, the original data is retrieved correctly by calculating $P_{i} = D_{(C_i)_c}$.

V. Security Analysis

The security of LACMMAA is analyzed through correctness, resistance to attacks, and cryptographic assumptions.

Theorem 1: If the DDH assumption holds, the overall construction is secured.

Let the adversary $\mathcal{A}$, with a non-negligible advantage $\mathcal{E} > 0$, can query for a secret key with a limitation that the obtained secret key cannot decrypt the challenge ciphertext. Under this constraint, the secure game of the multi-authority scheme and single authority scheme can be considered equivalent. Here, the DDH assumption can be challenged by building a challenger $\mathcal{C}$.

Initialization: $\mathcal{A}$ selects the access structure $(\mathbb{A}, \rho)$ to send it to $\mathcal{C}$ as a DDH challenge.
Setup: $\mathcal{C}$ performs the *Authority_Setup* to generate the public parameters. Then, it selects two integers $y_i, k_i \in z_r$ randomly and calculates the PK as $\{y_i . G, k_i . G\}$ for attribute i.
Phase 1: $\mathcal{A}$ sends G and attribute set to $\mathcal{C}$ to obtain the private key. Since all the obtained private keys cannot satisfy the access structure, $\mathcal{C}$ selects an integer $p \in z_r$ randomly and calculates the secret key of the attribute i as $U'_{S_i} = y_i + H(G_{i}) . k_i + p$.
Challenge: $\mathcal{A}$ chooses two messages, i.e., P_0 and P_1, of equal length, and sends them to $\mathcal{C}$. $\mathcal{C}$ randomly selects two vectors $v = (s, v_1, \dots, v_m)$ and $u = (0, u_1, \dots, u_m)$, and calculates $\gamma_x = \mathbb{A}_x . v$ and $\omega_x = \mathbb{A}_x . u$. Then, $\mathcal{C}$ selects a bit $\mathcal{B} \in \{0, 1\}$ and calculates $C_0 = P_{\mathcal{B}} + s . G$. This bit is used to specify the PT from the challenger, and if the adversary can guess this bit correctly, it wins the game. Next, $\mathcal{C}$ calculates the challenge ciphertext and sends it to $\mathcal{A}$.

$$C_{1,x} = \gamma_x . G + y_{\rho(x)} . G, C_{2,x} = \omega_x . G + k_{\rho(x)} . G$$

Phase 2: $\mathcal{A}$ can continue sending G and attribute set to $\mathcal{C}$.
Guess: $\mathcal{A}$ sends guess bit $\mathcal{B}_1 \in \{0, 1\}$ to $\mathcal{C}$. If $\mathcal{B}_1 = \mathcal{B}$, $\mathcal{C}$ outputs 0, which means that $\mathcal{A}$ has won the game; else, the output is 1. According to Theorem 1, the advantage of $\mathcal{A}$ is $\mathcal{E}$ and the probability that $\mathcal{A}$ guesses $\mathcal{B}_1 = \mathcal{B}$ correctly is $P(\mathcal{B}_1 = \mathcal{B}) = \frac{1}{2} + \mathcal{E}$. Thus, $P(\mathcal{B}_1 \neq \mathcal{B}) = \frac{1}{2}$. $\mathcal{C}$'s advantage to break the security game is $\mathcal{C} = \frac{1}{2}[P(\mathcal{B}_1 = \mathcal{B}) + P(\mathcal{B}_1 \neq \mathcal{B})] - \frac{1}{2} = \frac{\mathcal{E}}{2}$.

If the advantage of $\mathcal{A}$ is non-negligible, then, the advantage of $\mathcal{C}$ is also non-negligible. From this, it can be said that the proposed scheme is secured under the DDH assumption.

A. Correctness

The correctness of any cryptographic scheme relies on the cryptographic assumptions.

1) On receiving $U'_{s_i} = y_i + H(G^{k_i}).k_i + p$, α, C_C, and c_i, the DUA obtains C_D from C_C. If α satisfies $(\mathbb{A}, \rho)$, the DUA computes one constant set $\{c_x \in z_r\}$ in a polynomial time such that $\sum_{x\in X} c_x.\mathbb{A}_x = (1,0,\dots,0) = \varepsilon$. Here, $v = (s, v_1, \dots, v_m)$, $u = (0, u_1, \dots, u_m)$, $\lambda_x = \mathbb{A}_x.v$, and $\omega_x = \mathbb{A}_x.u$. So, $\sum_{x\in X} c_x.\lambda_x = s$ and $\sum_{x\in X} c_x.\omega_x = 0$. Then, the DUA calculates D_x as follows:

$$D_x = C_{1,x} - U_s.G + H(G^{k_i}).C_{2,x}$$
$$= \lambda_x.G + H(G^{k_i}).\omega_x.G - p.G$$

Again, the DUA calculates N_1 and N_2.

2) After receiving the partially decrypted ciphertext from the DUA, the DU calculates $c_i' = C_0 - N_1 - p.N_2 = c_i + s.G - s.G + p\sum_{x\in X} c_x.G - p\sum_{x\in X} c_x.G = c_i$. Then, the DU calculates $H(C^{k_i}).Q_D$. The CT is accepted only if $C_H == H(C^{k_i}).Q_D$. Finally, the DU calculates $P = D_{(C^{k_i})_C}$. Thus, the original data is retrieved correctly.

B. Data Confidentiality

A user without the sufficient attributes cannot generate N_1 and N_2. As a result, s/he cannot obtain c_i from $C_0 = c_i + s.G$, as well as the P from the C. Moreover, the message cannot be decrypted without all the partial secret keys $U_{s_i} = y_i + H(G^{k_i}).k_i$. Here, each A_i generates a portion of the secret key, and the CSP selects the re-encryption key. So, attribute authority, central authority, and CSP cannot obtain the original message. The DUA also cannot fully decrypt the CT unless it has DU's private key component p.

C. Collusion Attack

In the proposed scheme, if multiple users collude, they cannot be able to obtain the original message until at least one of them can successfully decrypt the message on their own. This is because of the GID associated with the U_s of each user, which makes it impossible to combine the partial secret keys in the decryption phase. For example, two users Y and Z intend to collude with each other with an aim to decrypt a message encrypted under the access policy $((C \wedge D) \vee A \wedge B)$. User Y only has the property C, and Z has the properties A and B. So, none of them is able to decrypt the ciphertext alone. Consider, Y and Z collude with each other. Then, Y receives $\lambda_x.G + H(G^{k_i})_Y.\omega_x.G$ and Z receives $\lambda_x.G + H(G^{k_i})_Z.\omega_x.G$ from the AA. Since $G^{k_i}{}_Y \neq G^{k_i}{}_Z$, Y and Z cannot collude to obtain $\{c_x \in z_r\}$, and $\sum_{x\in X} c_x.\mathbb{A}_x = (1,0,\dots,0) = \varepsilon$ is not true in this case. Thus, the proposed system can resist the collusion attack.

D. Key-Escrow Free

In any attribute-based encryption scheme, if an authority generates all the public and private keys and the authority is fully trusted, then, the authority is able to decrypt the encrypted data by using these keys. The proposed scheme needs all the users' secret keys generated by multiple attribute authorities, as well as the secret key of the CSP to successfully decrypt the ciphertext, so the proposed scheme is key-escrow-free.

E. Man-in-the-Middle Attack and Forgery Attack

In the proposed scheme, the DO computes the hash of the C as $C_H = H(C).D_D.G$. Any DU can verify the originality of the C by calculating $C_H == H(C).Q_D$, where $Q_D = D_D.G$. Even if hackers forge or intercept the communication to get any C, they are not able to calculate the respective C_H correctly as the DO's private key D_D is unknown to them. In any case, if a hacker is able to alter the C, C_H cannot be verified successfully by the DU. Thus, the C cannot be tampered or forged by illegal users and the proposed scheme can resist man-in-the-middle attacks and forgery attacks.

VI. Performance Analysis

The proposed scheme's performance is represented in the following subsections in comparison to other existing schemes [9, 15, 19, 20, 21, 22]. At first, healthcare data are collected by using IoT devices and gateways. The experiments are carried out to compute the time incurred for key generating, encryption, decryption, and total execution. In any cryptosystem, the key generation time, data encryption time, data decryption time, and execution time consumed by the system are measured to analyze the efficiency of the entire system. If the values of these parameters increase, a cryptosystem may perform slowly.

A. Experimental Environment

All the algorithms of the proposed LACMMAA are implemented by using the Java programming language. Here, the Java-based ECC library is used for underlying cryptographic operations. The experiments are conducted on an HP Pro 200 G4 Desktop with the configuration of Intel Core i5-10210U, 8 GB RAM, 2 TB HDD, and Windows 10 OS.

B. Security Features

The security features of the proposed scheme in comparison to other existing schemes are shown in Table II. The schemes proposed in [19, 20, 21] are not key-escrow free, and [9, 19, 20, 21] are ECC-based and pairing free. In [9, 15, 20], a single central or attribute authority is used for key generation. The proposed scheme is a pairing-free ECC-based system, which has multiple attribute authorities and an outsourced decryption unit and does not face the key-escrow problem.

TABLE II
Security Feature Comparison

Scheme	ECC	Key-escrow free	Multiple authority	Decryption outsourcing
[9]	𝕐	𝕐	ℕ	𝕐
[15]	ℕ	𝕐	ℕ	ℕ
[19]	𝕐	ℕ	𝕐	ℕ
[20]	𝕐	ℕ	ℕ	ℕ
[21]	𝕐	ℕ	𝕐	𝕐
Proposed scheme	𝕐	𝕐	𝕐	𝕐

C. Computational Cost

For any cryptographic system, the computational cost mainly depends on the system setup, key generation, encryption, and decryption processes. These processes may consist of many operations, such as hash, exponential, ECC-based scalar

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3167842, IEEE Transactions on Industrial Informatics

multiplication, bilinear pairing, etc. Among these operations, the most expensive is bilinear pairing, and scalar multiplication based on ECC is the least expensive operation [20]. 20 ECC-based scalar multiplications can be assumed as one bilinear pairing operation [9]. In this way, the point scalar multiplication ($\mathcal{M}$) can be taken as the unit of computational cost in ABE schemes. In Table III, the computational cost of the proposed scheme and other existing schemes is shown, where T_A is the total no. of attributes, T_L is the total no. of leaf nodes in the access structure. T_R is the total no. of attributes of the receiver, and $\mathcal{M}$ is one scalar multiplication. From Table III, it can be noticed the encryption time of the proposed scheme is slightly higher than [20, 21] that are based on KP-ABE. In [20, 21], the ciphertext is simply generated from an attribute set, so the time incurred for encryption is smaller than the proposed scheme. The overall setup overhead in the proposed scheme is a little high. As the system involves multiple attribute authority, the workload is distributed among all authorities and the setup overhead on each authority is reduced. Further, the decryption overhead of the end-users is reduced expressively by partially outsourcing the decryption process. The end-user performs only one scalar multiplication in order to complete the decryption process, so the overhead at the end-user is $(1)\mathcal{M}$ only.

D. Communication Cost

Communication cost is mainly dependent on the size of the message that needs to be transmitted. In Table IV, a comparison of the size of the private key, public key, and ciphertext of the proposed scheme and other similar schemes is shown. For simplicity, these comparisons are done at the same security level of 160-bit, and 160-bit ECC-based operation is considered as the unit of comparison and it is represented by S. Accordingly, the size of an ECC point is considered as $2S$. In the same way, the size of the access tree and attribute set is considered as S, and the size of the public key and the private key is considered as $2S$ and S, respectively.

To show the proposed scheme's effectiveness, the proposed LACMMAA, LEABE [20], EECPABE [22], and EBAC [21] are implemented by using an elliptic curve of size 512-bit finite field to achieve an 80-bit security level. The order of the elliptic curve group is 160-bit. Fig. 2(a) shows the computing time incurred for a key generation for different numbers of attributes. The proposed LACMMAA takes very less time for the key generation as compared to LEABE and EBAC. This is because these schemes attach an access policy with the secret key, which

TABLE III
COMPUTATIONAL COST COMPARISON

Scheme	Setup	Encryption	Pre-decryption	Decryption
[9]	$2\mathcal{M}$	$(4T_L + 1)\mathcal{M}$	$(T_R)\mathcal{M}$	$(2)\mathcal{M}$
[19]	$(T_A + 1)\mathcal{M}$	$(4T_L + 1)\mathcal{M}$	–	$(T_R + 1)\mathcal{M}$
[20]	$(T_A + 1)\mathcal{M}$	$(T_L + 1)\mathcal{M}$	–	$(T_R + 1)\mathcal{M}$
[21]	-	$(T_L + 1)\mathcal{M}$	$(T_R + 1)\mathcal{M}$	$(3)\mathcal{M}$
[22]	-	$(4T_A + 1)\mathcal{M}$	$(T_R + 1)\mathcal{M}$	$(1)\mathcal{M}$
Proposed scheme	$(2T_A + 1)\mathcal{M}$	$(3T_L + 1)\mathcal{M}$	$(T_R)\mathcal{M}$	$(1)\mathcal{M}$

TABLE IV
COMMUNICATION COST COMPARISON

Scheme	Private key	Public key	Ciphertext
[9]	$(2T_R + 4)S$	$(T_A + 4)S$	$(2T_L + 6)S$
[19]	$(T_R)S$	$(2T_A + 2)S$	$(6T_L)S$
[20]	$(T_L + 1)S$	$(2T_A + 2)S$	$(2T_L + 2)S$
[21]	$(2T_R + 4)S$	$(2T_A + 4)S$	$(2T_L + 2)S$
[22]	$(T_R)S$	$(2T_A + 2)S$	$(3T_L + 1)S$
Proposed scheme	$(T_R)S$	$(2T_A + 1)S$	$(2T_L + 1)S$

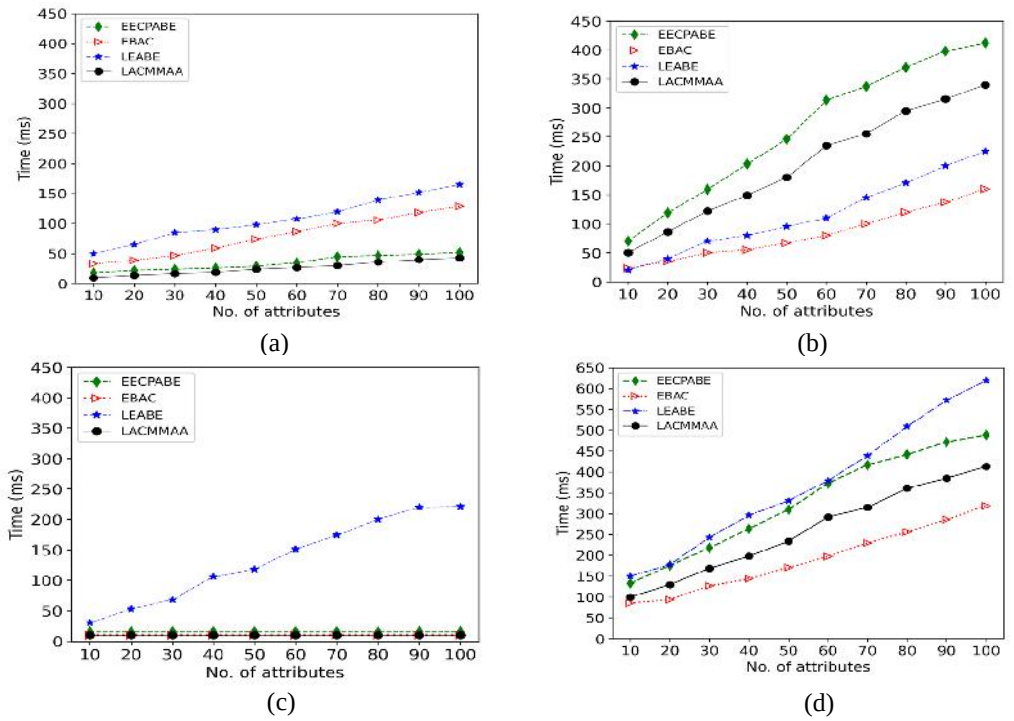

Fig. 2. Time incurred by the proposed scheme for different number of attributes. (a) Key generation time (b) Encryption time (c) Decryption time, and (d) Total execution time.

increases the key generation time. Fig. 2(b) shows the encryption time for different numbers of attributes. LACMMAA takes less time for encryption as compared to EECPABE scheme, however, more time than LEABE and EBAC. In EECPABE, the encryption time increases linearly with the increasing number of attributes. LEABE and EBAC are based on KP-ABE and take less time for data encryption because they add an access policy to the key rather than the ciphertext. Similarly, Fig. 2(c) shows the computing time occurred in decryption for different numbers of attributes. The decryption time in LACMMAA is less as the DU performs only one scalar multiplication for the decryption. It can be noticed that LACMMAA minimizes the decryption overhead significantly. The decryption at the end user's side takes minimum time since most of the complex calculations are performed at the data user assistant's side. However, LEABE performs the entire decryption operation locally, which makes the decryption time high. At last, Fig. 2(d) shows the total execution time of the proposed scheme for different numbers of attributes. Here, the sum of the execution time incurred for all the algorithms is considered as the total execution time.

VII. Conclusions and Future Work

In this paper, a lightweight fine-grained access control model is proposed by using ECC and CP-ABE schemes for IoT-enabled healthcare systems. In the proposed scheme, multiple attribute authorities manage the user attributes to reduce the workload of the central authority. Thus, the proposed scheme also solves the key-escrow problem. Here, the computational overhead of the end-users is reduced by outsourcing the decryption process to a data user assistant. The security analysis proves that the proposed scheme is protected against collusion attacks, man-in-the-middle attacks, and forgery attacks. Furthermore, the performance analysis shows the efficiency of the proposed scheme over some existing schemes. The proposed scheme can be used in different fields like big data, smart agriculture, and many more, where access control is a major necessity. However, in this work, the attribute revocation problem has not been solved. In the future, this proposed work can be extended to solve the attribute revocation problem.

References

[1] R. W. L. Coutinho and A. Boukerche, "Modeling and analysis of a shared edge caching system for connected cars and industrial IoT-based applications", *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2003-2012, 2020.

[2] S. Suganya, "Enhancing security for storage services in cloud computing", *IEEE Current Trends in Engineering and Technology*, vol. 3, no. 6, pp. 283-287, 2014.

[3] S. Namasudra, "Fast and secure data accessing by using DNA computing for the cloud environment", *IEEE Transactions on Services Computing*, 2020. DOI: 10.1109/TSC.2020.3046471

[4] S. H. Islam et al., "Provably secure identity-based signcryption scheme for crowdsourced industrial internet of things environments", *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2904-2914, 2017.

[5] G. Y. Liu et al., "Secure and fine-grained access control on e-healthcare records in mobile cloud computing", *Future Generation Computer Systems*, vol. 78, pp. 1020-1026, 2018.

[6] S. Namasudra et al., "Blockchain-based medical certificate generation and verification for IoT-based healthcare systems", *IEEE Consumer Electronics Magazine*, 2022. DOI: 10.1109/MCE.2021.3140048

[7] H. Deng et al., "A flexible privacy-preserving data sharing scheme in cloud-assisted IoT", *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11601-11611, 2020.

[8] K. Sowjanya and M. Dasgupta, "A ciphertext-policy attribute based encryption scheme for wireless body area networks based on ECC", *Journal of Information Security and Applications*, vol. 54, pp. 1-10, 2020.

[9] K. Sowjanya, M. Dasgupta, and S. Ray, "A lightweight key management scheme for key-escrow-free ECC-based CP-ABE for IoT healthcare systems", *Journal of Systems Architecture*, vol. 117, pp. 1-10, 2021.

[10] H. Deng et al., "Ciphertext-policy hierarchical attribute-based encryption with short ciphertexts", *Information Sciences*, vol. 275, pp. 370-384, 2017.

[11] L. Zhang et al., "Hidden ciphertext policy attribute-based encryption with fast decryption for personal health record system", *IEEE Access*, vol. 7, pp. 33202–33213, 2019.

[12] M. Chase and S. Chow, "Improving privacy and security in multi-authority attribute-based encryption", in *Proceedings of the 16th ACM conference on Computer and communications Security*, 2009, pp. 121-130.

[13] X. Yan et al., "Privacy-preserving multi-authority attribute-based encryption with dynamic policy updating in PHR", *Computer Science and Information Systems*, vol. 16, pp. 831–847, 2019.

[14] L. Zhang, Y. Ye, and Y. Mu, "Multiauthority access control with anonymous authentication for personal health record", *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 156-167, 2021.

[15] 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.

[16] A. Lewko et al., "Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption", in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2010, pp. 62–91.

[17] L. Li et al., "A ciphertext-policy attribute-based encryption based on an ordered binary decision diagram", *IEEE Access*, vol. 5, pp. 1137-1145, 2017.

[18] C. Hu et al., "Secure and efficient data communication protocol for wireless body area networks", *IEEE Transactions on Multi-Scale Computing Systems*, vol. 2, no. 2, pp. 94-107, 2017.

[19] 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.

[20] 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.

[21] X. Qin, Y. Huang, and X. Li, "An ECC-based access control scheme with lightweight decryption and conditional authentication for data sharing in vehicular networks", *Soft Computing*, vol. 24, pp. 18881-18891, 2020.

[22] R. Cheng et al., "An efficient ECC-based CP-ABE scheme for power IoT", *Processes*, vol. 9, pp. 1-16, 2021.

[23] G. Mohindru, K. Mondal, and H. Banka, "Different hybrid machine intelligence techniques for handling IoT-based imbalanced data", *CAAI Transactions on Intelligence Technology*, vol. 6, no. 4, pp. 405-416, 2021.

[24] W. Li et al., "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.

[25] R. M. Alguliyev, R. M. Aliguliyev, and L. V. Sukhostat, "Efficient algorithm for big data clustering on single machine", *CAAI Transactions on Intelligence Technology*, vol. 5, no. 1, pp. 9-14, 2020.