Received: June 7, 2023. Revised: August 14, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Cloud Security with Lightweight ABE on Mobile IoT Devices

Keerti Naregal[1]* **Vijay Kalmani[1]**

[1]*Department of Computer Science and Engineering, Jain College of Engineering, Belagavi, India*
* Corresponding author's Email: keertinaregal@gmail.com

Abstract: The internet of things (IoT) faces significant obstacles due to insufficient identity recognition and evolving network architecture, leading to concerns about the confidentiality of data and causing anxiety. The attribute-based encryption (ABE) techniques have recently been considered a solution to guarantee the security of data transfer and precise data sharing. However, most of the existing methods used the attribute-based encryption (ABE) technique, which requires a lot of computation power and is unsuitable for IoT devices with minimal resources. Researchers have achieved improvements in establishing practical methods for cloud security on mobile IoT devices using lightweight ABE. In this paper, the ciphertext policy-revocable and searchable attribute-based encryption (CP-RSABE) method is proposed to protect privacy and security. The proposed methods greatly lower the cost of computing IoT devices with the availability of multiple-keyword searchers for the users of data. The user's side of computation is very efficient, and the cloud server handles most of the computing tasks. The proposed method performs significantly better in terms of ciphertext size, decryption time, and parameter size. The method achieves data security, privacy preservation, and mobile terminal operations that are suitable for applications of IoT methods. The existing methods such as online/offline multi authority-ABE with cryptographic reverse firewalls (OO-MA-ABE-CRF), ciphertext policy ABE (CP-ABE), ABE with full privacy protection (ABE-FPP) are used to justify the effectiveness of CP-RSABE method. The proposed method CP-RSABE achieves the encryption time (0.0163s), decryption time (0.25s), communication overhead (3.4KB), size of secret key (5.1KB), and size of ciphertext (10.7KB) compared to the OO-MA-ABE-CRF, CP-ABE, ABE-FPP.

Keywords: Attribute-based encryption, Cloud computing, Fine-grained access control, Internet of things, Privacy protection.

1. Introduction

Industrial internet of things (IoT) is a new paradigm altering contemporary industries by associating numerous devices with gateways of IoT, base stations, and points of access, enabling ubiquitous data collection and exchange over the sectors of various industries. Local edge computing nodes gather a lot of data from different IoT devices, process it, and send it to distant cloud servers where it can be used by data consumers like assembly line workers, technicians, and business managers [1]. Cloud-based IoT solutions make using the same data across many services feasible, making it easier to store and process the collected data and allowing user mobility. However, as the cloud cannot be regarded as a trusted entity, outsourcing the acquired data to a server of the cloud increases concerns about precise access control and data confidentiality [2]. Privacy and security issues are problems that essential healthcare faces. Healthcare data kept on a third-party cloud service or transmitted in real-time are susceptible to numerous threats [3]. The medical industry has seen significant changes in e-health services as a result of the development of emerging technologies, particularly mobile cloud computing and the Internet of Medical Things [4]. In order to enable intelligent transportation services through data analysis, and mining numerous IoT devices produce or collect data that involves sensitive personal information, which is then stored in the cloud. In this situation, data owners are unable to handle their data directly, and the trustworthiness and reputation of cloud service providers (CSPs) are crucial for data protection [5]. Cloud providers handle security and privacy problems as a matter of urgent and high priority. Data privacy is one of the main security issues with data sharing. In addition, terminals of users are frequently resource-constrained mobile devices with tiny storage space and low processing speed [6]. Low latency and real-time processing are essential components of industrial applications that cloud computing insufficiently supports [7]. To address these issues, the fog computing paradigm developed as a complementary technology to cloud computing [8].

Data encryption and storage on the cloud server are the solutions to this issue. attribute-based encryption (ABE), which also ensures control of fine-grained access over data, can be used to do this. Because only open parameters are utilized for encryption in ABE, anyone can encrypt. Instead, decryption is carried out using a decryption key that is unique to each user and created by trusted third party [9]. Short ciphertext ABE algorithms have been proposed by researchers, and some of them can reduce the computing cost required for decryption. The selective security model however requires attackers to disclose the access structure to be targeted before the setup process, and this model has demonstrated that these schemes are secure [10]. Before the standard ABE can be used in reality, however, a number of constraints must be resolved. The considerable decryption overhead of ABE is one of its drawbacks. Since the IoT ecosystem for lightweight devices has several limitations in terms of processing resources, cost of bandwidth, and battery life, ABE cannot be applied directly to healthcare IoT networks [11]. A highly promising method for achieving one-to-many encryption and access control of fine-grained across encrypted data is ciphertext policy attribute-based encryption (CP-ABE). The secret key is connected to the user set of attributes in the CP-ABE scheme while the policy of access is related to the ciphertext. Data users were able to access the data if the set of user attributes included in the secret keys fits the policy of access for the ciphertext [12]. However, the access policy is explicitly added to the data ciphertext in schemes of conventional CP-ABE which means that anyone who obtains the ciphertext, including the provider of a cloud server, can deduce some secret information about the data content or the recipients of secret data from the policy [13]. In the following, the primary contribution of this paper is summarized as follows:

- Lightweight encryption mechanism: The data owners can encrypt their data by executing lightweight computations, and the cloud server handles all activities of expensive computational during the phase of encryption without learning anything about the file of underlying data.
- Flexible and scalable delegation of key and user revocation mechanism: The phases of key delegation and user revocation are individually handled by each of the key generator authorities and users are free to request their secret keys from whatever they choose the key authority. Additionally, the primary authority adds additional key generator authorities whether the system requires more computational power.
- Security definition and proof: The system model and security definition of the proposed scheme are formalized. In addition, demonstrate that our system is secure in the conventional model.

The IoT ecosystem for lightweight devices has different limitations in terms of processing resources, bandwidth cost, battery life, and ABE cannot be directly used in healthcare IoT networks due to computational requirements. In this proposed paper, the new model is developed to overcome the privacy and security issues in an existing model for future work.

The rest of this research is organized as follows. Section 2 discusses the literature survey. Section 3 describes the proposed methodology. Section 4 describes the results. The conclusion of this research work is given in section 5.

2. Literature survey

Zhang [14] implemented a light searchable attribute-based encryption method (LSABE) to adopt the search of muti-keyword in an environment of distributed Internet of Things with the provision of key management decentralization. To efficiently create and maintain the secret/public keys in the distributed IOT context, LSABE method was extended namely LSABE-MA to the settings of multi-authority. LSABE performs significantly better in terms of ciphertext size, decryption time, and parameter size. The proposed method has the drawback of private information about users saved on a cloud server being disclosed or misused.

Juyan Li [15] presented an online/offline multi authority-ABE with cryptographic reverse firewalls (OO-MA-ABE-CRF) method for data privacy security in IoT and employs multi-authority technology to further decrease dependence on a single authority. CRF technology to fully protect user security and privacy from leakage information, online/offline, and outsourcing decryption technology to lower user computing costs and storage. This method has achieved fine-grained access control over data as well as data security protection. However, network latency occurs by the continual communication required for online/offline MA-CP-ABE with the central authority and attribute authority.

Sowjanya [16] implemented a lightweight ciphertext policy attribute-based encryption (CP-ABE) key management method using elliptic curve cryptography (ECC) to ensure that the internet of things-based system of healthcare has fine-grained access control. The developed management of key technique in the CP-ABE is free of key-escrow and considerably minimizes the data receiver's decryption overhead and existing ABE methods are based on bilinear pairing. The CP-ABE method is more effective in terms of features security, communication overheads, and computing. CP-ABE has drawbacks in terms of the attributes of managing users and setting policies.

Tian [17] presented a lightweight attribute-based encryption with full privacy protection (ABE-FPP) method that achieves protection of full privacy in three essential stages: control access, generation of key, and partial decryption. Hybrid-verification technique was implemented to safeguard privacy during partial decryption. The ABE-FPP method increases efficiency for devices with limited resources and improves computation speed during the encryption and decryption stages while protecting user privacy and security. Compared to other schemes the proposed method has a shorter ciphertext length.

Ahmed Saidi [18] implemented a novel collaborative method SHARE-ABE for maintaining privacy based on CP-ABE and the most time-consuming decryption tasks were assigned to fog nodes using fog computing. SHARE-ABE enables the reduction of networking and computational decryption costs for resource-constrained IoT systems by using a new chained collaboration method among various fogs. Especially for IoT devices with resource-constrained, the implemented method achieves security and efficiency. However, the method was challenging to maintain coordination and synchronization across numerous fog nodes.

Aghili, S.F. [19] presented a multi-level security attribute based access control method (MLS-ABAC) depending on user attributes that are both static and dynamic as well as the security level that has been allocated to satisfy the NIST ABAC requirements model. With a constant decryption key size, the ABE structure was based on CP-ABE. The privacy and security of IoT systems were increased through the addition of a security level check before partial decryption. However, the MLS-ABAC method needs the storage of quasi-constant.

Fugkeaw [20] implemented lightweight medical devices (LightMED) to enable outsourced IoT-EMR's secure, lightweight access control with the use of blockchain technology and fog computing. To allow control authentication of decentralized access and auditing, the method makes use of blockchain technology and contains the secure aggregation of healthcare data based on the data encryption of IoT and other treatment records. The method produces the lowest processing costs for encryption and decryption on the devices at the end-user indicating improved efficiency. However, this method determines half of the issue because this method does not address the characteristics visibility in the ciphertext.

The disadvantage of personal user data stored on a cloud server being revealed or used, characteristics of managing users and setting policies, and forgery attack or replacement assault occurring are some of the limitations for cloud security with lightweight ABE on mobile IoT devices that were mentioned above.

3. Proposed methodology

The existing method has a drawback in terms of the attributes of managing users and setting policies and to encrypt data, the data owner needs to utilize the public key of each authorized user. To overcome privacy and security issues, the ciphertext policy-revocable and searchable attribute-based encryption (CP-RSABE) method is proposed.

System model:

Central authority (CA), cloud service provider (CSP), domain authorities (DAs), various data users, and data owners (DO) are the five entities of the system model. The overview of cloud security with lightweight ABE on mobile IoT device block diagram is shown in Fig. 1.

3.1 Data owner (DO)

In IoT networks, DO simulates small devices and small sensors. It is responsible for creating and encrypting the shared data, defining the structure of access, and dividing the data encrypted into blocks. To transmit or upload the data to the storage of cloud, the data owner must be an active user of the system. They have constrained processing and storage capacities. They gather environmental data and encrypt it in accordance with an access control

Received: June 7, 2023. Revised: August 14, 2023.

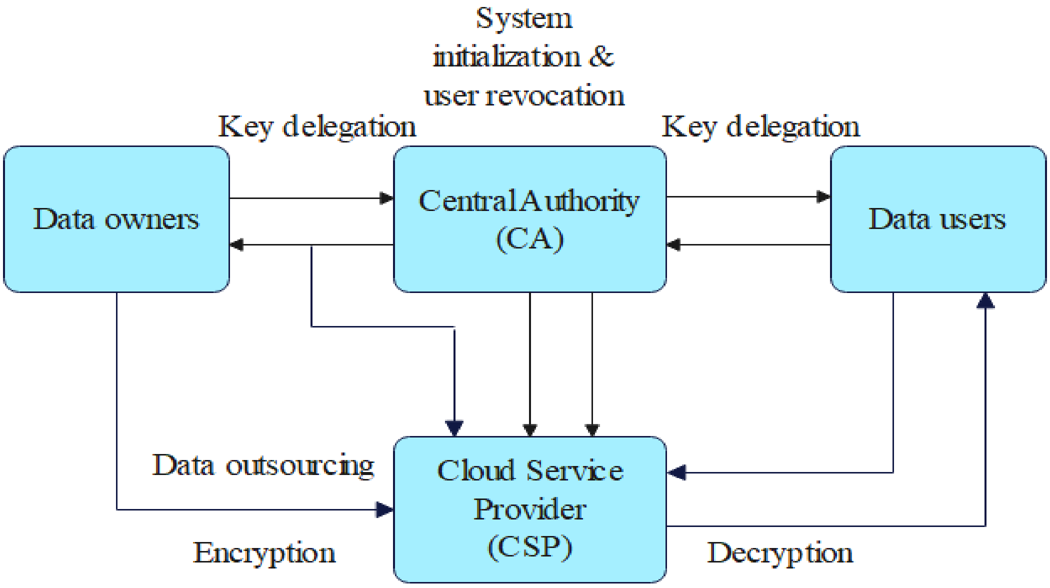

Figure. 1 Block diagram for the proposed method

strategy. The CSP receives the encrypted data as a service. Before being sent to the cloud, the data is encrypted using this access policy. For the designated receivers, the access policy is defined by the owner of the data and it is used for encryption.

3.2 Central authority (CA)

CA is a completely trustworthy and honest entity that handles user registration for cloud storage and creates each data user's private key by interacting with the user. The key algorithms that were executed by this entity are setup and keygen. An easy-to-use, efficient method for creating and storing asymmetric key pairs is provided by a server that requires the usage of public key infrastructure such as for encrypting, decrypting, validating, or signing. System parameters generation and DA initialization are responsible. The DA are in charge of creating data user's secret keys based on their attributes and revoking them when they are missing any of those attributes. Each DA offers user revocation and key delegation services connected to certain attributes.

3.3 Cloud service provider (CSP)

The amount of storage and processing power available to the CSP is virtually limitless. For data users and data owners, it offers computational and storage services. The most computationally intensive tasks are given by a data owner who wants to encrypt their data without disclosing any partial data. Additionally, it can offer computational services to data users. In fact, it can handle the majority of the costly tasks associated with decryption so that no information regarding the secret keys of data user's and supporting files of data are revealed. It serves as a data center and offers a service of data sharing. Due to the assumption that this entity is also semi-trusted like key delegation, the cloud server also produces secret keys for users (depending on their sets of attributes) using their own master secret keys.

3.4 Domain authority (DA)

DA are in charge of creating data user's secret keys based on their attributes and revoking them when they are missing any of those attributes. Each DA offers user revocation and delegation of key services connected to certain attributes. When a user of data has access to an attribute, it chooses any DA that supports the attribute and request that to produce the appropriate secret key. It determines whether or not the user of data processes the attribute. If so, it gives the data user the private key with the secret key. Additionally, it can assist owners of data in producing their public and secret keys.

3.5 Data users

Data users acquired a private key whose attributes establish with the access structure as a cloud user. Data users are always required to safeguard the data they handle in accordance with this policy and any relevant data security guidelines. Data users aim to use the files of outsourced data for a variety of users, including, research marketing, etc. Data users can ask the CSP for partial decryption using their attribute secret keys. After that, users can recover their desired data files using lightweight operations. Each data user receives various authorities in accordance with the attributes and has a user identifier assigned to them by the CA. The set of keywords can be used by data users to search the ciphertexts.

3.6 System initialization

System initialization is managed by the CA. It begins by creating a workload resulting from user revocation and key delegation operations, it initializes several DA. The CA first chooses a security parameter λ and a universal attribute set U .

It produces the Master Secret-Key MSK and public parameter params by running

$$(params, MSK) \leftarrow setup(1^{\lambda}, U) \tag{1}$$

It generates public keys and master secret-keys of DA by running

$$\{PK_j, MSK_j\}_{j=1}^{n} \leftarrow KAGen(params, MSK, \{id_j\}_{j=1}^{n}, \{U_j\}_{j=1}^{n}) \tag{2}$$

The two aforementioned algorithms Eqs. (1) and (2) are described below:

$\boldsymbol{setup(1^{\lambda}, U)}$*:* When the universal attribute set U and the security parameter is input,

$$params = (\lambda, q, G1, G2, \hat{e}, g1, g2, g3, h1, h2, \{pk_i\}_{i\epsilon U}, P1, P2, \pi_S, H) \tag{3}$$

Where $params$ – global public parameter and

$$MSK = (x, y, g4, msk_1, msk_2\ \{s_i\}_{i\epsilon U}) \tag{4}$$

Where MSK – Master Secret Key of the CA.

By combining the Eqs. (3) and (4), it produces (5), (6).

$\boldsymbol{KAGen}\ (params, MSK, \{id_j\}_{j=1}^{n}, \{U_j\}_{j=1}^{n})$ It first chooses $s_j' \leftarrow z_q$ and evaluates,

$$SK_j = msk_1 g_4 id_j^{s_j'} \tag{5}$$

And

$$PK_j = msk_2 g_4 id_j^{s_j'} \tag{6}$$

Where sk, pk– secret key, public key
U - Universal attribute key
λ - Security parameter
H - Hash function
π_S - Symmetric key

For $j = 1, , n$. Then, it returns $\{PK_j, MSK_j\}_{j=1}^{n} = 1$, where $MSK_j = (SK_j, \{sj\}_{i\epsilon U_j})$.

3.7 Key delegation

Key delegation is run by DA. They create secret keys of data users in this key delegation based on their qualities. They also give data owners access to their public and secret keys. If a user of data has an attribute with the value $i\epsilon U$, it should request that the DA that supports the attribute to produce the attribute's secret key. A DA with the identifier id_j first determines if the data user has the attribute i when it receives a request (i, id_u) from the user of the data. If not, it aborts. Otherwise, it performs Eqs. (7) and (8).

$$sk_{i,u}^{(j)} \leftarrow User.KeyGen\ (params, MSK_j, id_{u,}, i) \tag{7}$$

and provides data users with $sk_{i.u}^{(j)}$. Additionally, DA runs.

$$(sk_o, pk_o) \leftarrow Owner.KeyGen\ (params) \tag{8}$$

When an owner of data requests that to produce their secret-key and public-key and provides the data owner with sk_o. Also. it delegated pk_o to the CSP.

$\boldsymbol{User.KeyGen}(params, MSK_j, id_u, i)$: Given $MSK_j = (sk_j, \{s_i\}_{i\epsilon U_j})$ and an attribute i , this algorithm determines if $i\epsilon U_j$ or not. If so, it gives back Eq. (9).

$$sk_{i,u}^{(j)} = sk_j id_u^{s_i} \tag{9}$$

If not, it displays a message of error ⊥.

$\boldsymbol{Owner.KeyGen}(params)$: First it chooses $(sk_o, \leftarrow z_q)$ and produces Eqs. (10) and (11)

$$pk_o^{(1)} = p_2^{sk_o}, pk_o^{(2)} = g_1^{sk_o}, pk_o^{(3)} = g_3^{sk_o}, and\ pk_{i,o} = pk_i^{sko} \tag{10}$$

It returns (sk_o, pk_0), where

$$pk_0 = (\ pk_o^{(1)}, pk_o^{(2)}, pk_o^{(3)}, \{pk_{i,o}\}_{i\epsilon U}) \tag{11}$$

Where sk_o - data owner's secret key
pk_0 – data owner public key
id_j - identifier

3.8 Data outsourcing

Data outsourcing is managed by CSP and the data owners. When an owner of data wants to distribute the information, the owner first creates a policy of access control. Then, utilizing the CSP's computing capacity, the owner encrypts the data that is covered by the policy of access control. Asymmetric encryption like rivest shamir adleman (RSA) uses two dissimilar connected keys. Both the private and public keys encrypted the message using RSA cryptography. A message is decrypted using the opposite key to that which was used to encrypt it. Each RSA user has a pair of keys made up of their private and public keys. The private key must be protected as the term implies. For long-term archival and analysis of online\offline, the CSP stores the encrypted data file. A data owner owns a secret key sk_o and data file M encrypts their data at this phase, then stores the encrypted data in the CSP. It initially prevents authorized data users from accessing its data by defining access tree $\mathcal{T}$. After that, it executes -

$$PCT_{\mathcal{T}} \leftarrow Owner.Enc(params, sk_o, M, \mathcal{T}) \quad (12)$$

Returning a CSP a partial ciphertext $PCT_{\mathcal{T}}$. The CSP generates a ciphertext

$$CT_{\mathcal{T}} \leftarrow CSP.Enc(params, pk_o, PCT_{\mathcal{T}}) \quad (13)$$

Using the data owner's public-key pk_o. The two aforementioned algorithms Eqs. (12), (13) are expressed as below in Eqs. (16), (21)

$\boldsymbol{Owner.Enc}$ $(params, sk_0, M, \mathcal{T})$: It chooses $r \leftarrow z_q$ and evaluates $r' = r - sk_o$ and produces Eq. (14).

$$k = H(P_1^r) \quad (14)$$

After that, it runs Eq. (15)

$$\{q_{v_i}(0)\}_{v_{i\epsilon L_{\mathcal{T}}}} \leftarrow share_q(\mathcal{T}, r') \quad \text{and} \quad C \leftarrow Enc(M, K). \quad (15)$$

The partial ciphertext is then produced

$$PCT_{\mathcal{T}} = (C, r', \mathcal{T}, \{q_{v_i}(0)\}_{v_{i\epsilon L_{\mathcal{T}}}}) \quad (16)$$

$\boldsymbol{CSP.Enc}$ $(params, pk_0, PCT_{\mathcal{T}})$: Given $pk_o = \left(pk_o^{(1)}, pk_o^{(2)}, pk_o^{(3)}, \{pk_{i.o}\}_{i\epsilon U}\right)$ and $PCT_{\mathcal{T}} = (C, r', \mathcal{T}, \{q_{v_i}\}_{v_{i\epsilon L_{\mathcal{T}}}})$, this technique calculates C', C'', C_i, C_i' in Eqs. (17), (18), (19), and (20)

$$C' = g_3^{r'} pk_0^{(3)} = g_3^r, \quad (17)$$

$$C'' = P_2^{r'} pk_o^{(1)} = p_2^r, \quad (18)$$

and for each $v_i \epsilon L_{\mathcal{T}}$, it sets

$$C_i = g_1^{q_{v_i}(0)} pk_0^{(2)} = g_1^{q_{v_i}(0)+sk_0} \quad (19)$$

$$C_i' = (pk_{ig_3}^{-1})^{q_{v_i}(0)} pk_{i,o} (pk_o^{(3)})^{-1} = (pk_{ig_3}^{-1})^{q_{v_i}(0)+sk_0} \quad (20)$$

It returns

$$CT_{\mathcal{T}} = (\mathcal{T}, C, C', C'', \{C_i\}_{v_{i\epsilon L_{\mathcal{T}}}}, \{C_i'\}_{v_{i\epsilon L_{\mathcal{T}}}}) \quad (21)$$

where $\mathcal{T}$ - Access tree
$CT_{\mathcal{T}}$ - Ciphertext
$PCT_{\mathcal{T}}$ - partial ciphertext

3.9 User revocation

User revocation is managed by CSP and DA. DA updates the public and secret system parameters linked to the attribute when a data user overlooks it. The CSP then re-encrypts without knowing anything about the files of outsourced data and secret parameters, the outsourced files of data by the new parameters. Finally, DA update the secret keys for the unauthorized users of data that relate to the attribute. DA first runs and gets back a new secret parameter $\widetilde{s_{\iota_0}}$, a new public parameter $\widetilde{pk_{\iota_0}}$, and an updated key $UKey_{i_0}$. The remaining DA are given to $\widetilde{s_{\iota_0}}$, $\widetilde{pk_{\iota_0}}$ is published to the system, and the DA, CSP are given to $UKey_{i_0}$. The CSP re-encrypts $CT_{\mathcal{T}}$ to $\widetilde{CT_{\mathcal{T}}}$ by performing for each ciphertext $CT_{\mathcal{T}}$ that has a leaf node linked to i_0. Additionally, when a DA is asked to update its secret key $sk_{\iota_0,u}^{(J)}$ by an authorized data user with the identifier id_u, the DA runs and gets back $\widetilde{sk_{\iota_0,u}^{(J)}}$ to the user of data. The aforementioned algorithms in Eqs. (22), (23), and (24) are described below:

$$\left(\widetilde{s_{\iota_0}}, \widetilde{pk_{\iota_0}}, UKey_{i_0}\right) \leftarrow UpdateParams(params, MSK_J, i_0) \quad (22)$$

$$\widetilde{CT_{\mathcal{T}}} \leftarrow ReEnc(params, i_0, CT_{\mathcal{T}}, UKey_{i_0}) \quad (23)$$

$$\widetilde{sk_{\iota_0,u}^{(J)}} \leftarrow UpdateKey(params, id_u, i_0, sk_{i_0,u}^{(j)}, UKey_{i_0}) \quad (24)$$

$\boldsymbol{UpdateParams}(params, MSK_J, i_0)$: This technique checks if $i_0 \epsilon U_j$ or not. If not, it gives back $\perp$. Otherwise, produce Eq. (25) and gives back Eq. (26). First, it chooses $\widetilde{s_{\iota_0}} \leftarrow Z_q$ and evaluates $\widetilde{pk_{\iota_0}} =$

Received: June 7, 2023. Revised: August 14, 2023.

$g_1^{\widetilde{s_{\iota_0}}}$ and,

$$UKey_{i_0} = \widetilde{s_{\iota_0}} - s_{i_0} \tag{25}$$

$$\text{It gives back } \widetilde{s_{\iota_0}}, \widetilde{pk_{\iota_0}}, \text{ and } UKey_{i_0} \tag{26}$$

ReEnc$(params, i_0, CT_{\mathcal{T},} UKey_{i_0}\ )$: Given an attribute i_0 a ciphertext in Eq. (27).

$$CT_{\mathcal{T}} = (\mathcal{T}, C, C', C'', \{C_i\}_{v_{i \epsilon L_{\mathcal{T}}}}, \{C_i'\}_{v_{i \epsilon L_{\mathcal{T}}}}) \tag{27}$$

So that $\mathcal{T}$ has a node of leaf linked with i_0 and an update key $UKey_{i_0}$, this technique generates a ciphertext of re-encryption in Eqs. (28), (29).

$$\widetilde{CT_{\mathcal{T}}} = (\mathcal{T}, C, C', C'', \{C_i\}_{v_{i \epsilon L_{\mathcal{T}}}}, \{C_i'\}_{v_{i \epsilon L_{\mathcal{T}} \setminus \{v_{i_0}\}}} \cup \{\widetilde{C_{\iota_0}'}\}) \tag{28}$$

Where

$$\widetilde{C_{\iota_0}'} = \widetilde{C_{\iota_0}'} C_{i_0}^{UKey_{i_0}} \tag{29}$$

Update Key$(params, id_u, i_0, sk_{i_0,u}^{(j)}, UKey_{i_0})$: This algorithm modifies a secret-key $sk_{i_0,u}^{(j)}$ linked with an attribute i_0 as follows in Eq. (30):

$$\widetilde{sk_{\iota_0,u}^{(J)}} = sk_{i_0,u}^{(j)} id_u^{UKey_{i_0}} \tag{30}$$

where s_{i_0}, p_{i_0} - secret parameter, public parameter

$\widetilde{pk_{\iota_0}}, \widetilde{s_{\iota_0}}$ - new secret parameter, new public parameter.

$UKey_{i_0}$ - Update key

i_0 - attribute

$\widetilde{CT_{\mathcal{T}}}$ - Re-encrypted ciphertext

3.10 Decryption

Decryption is managed by the data users and CSP. The process of restoring an encrypted message to its original format is known as decryption. It uses the opposite key while encrypting and decrypting generic data with RSA. In this stage, eligible data users can acquire their required data by decrypting the outsource encrypted data files utilizing the processing resources of the CSP. A data user with the attribute set Att_u first determines whether or not a subset $S \subseteq Att_u$ meeting $\mathcal{T}$ exists before attempting to access external files of data matching to a ciphertext $CT_{\mathcal{T}}$. If not, it terminates. Otherwise, it examines the set of secret-key $\{sk_{i.u}^{(j_i)}\}_{i \epsilon S}$ and runs

$$(s, tk_u) \leftarrow Dec.TokenGen\left(params, id_u, \{sk_{i.u}^{(j_i)}\}_{i \epsilon S}\right) \tag{31}$$

The data user gives tk_u to the CSP while maintaining its confidentiality. The data user receives a partially decrypted copy of the ciphertext M' when the CSP executes

$$M' \leftarrow CSP.Dec(params, tk_u, CT_{\mathcal{T}}) \tag{32}$$

The data user then uses

$$M \leftarrow User.Dec(params, M', CT_{\mathcal{T}}, s) \tag{33}$$

The three aforementioned algorithm in Eqs. (31), (32), and (33) are described as below:

Dec.TokenGen$(params, id_u, \{sk_{i,u}^{(j_i)}\}_{i \epsilon S})$: This algorithm returns the secrets and a token of decryption after selecting $S \leftarrow Z_q$ in Eq. (34)

$$tk_u = \{t_u, tk_{i,u}, tk_{i,u}'\}_{i \epsilon S}, \tag{34}$$

Where,

$t_u = id_u^s, tk_{i,u} = (sk_{i,u}^{(j_i)})^s$, and $tk_{i,u}' = C_i^s$

CSP.Dec$(params, tk_u, CT_{\mathcal{T}})$: Given $tk_u = \{t_u, tk_{i,u}, tk_{i,u}'\}_{i \epsilon S}$ and $CT_{\mathcal{T}} = \left(\mathcal{T}, C, C', C'', \{C_i\}_{v_{i \epsilon L \mathcal{T}}}, \{C_i'\}_{v_{i \epsilon L_{\mathcal{T}}}}\right)$, this technique first calculates Eq. (35).

$$\frac{\hat{e}\left(tk_{i,u}, C_i\right)}{\hat{e}\left(PK_{j_i}, tk_{i.u}'\right)\hat{e}(t_u, C_i')} = P_1^{s(q_{v_i}(0)+sk_o)} P_2^{-s(q_{v_i}(0)+sk_0)} * \hat{e}(t_u, g_3)^{s(q_{v_i}(0)+sk_o)} \tag{35}$$

For each $i \epsilon S$. Next, by employing the technique of polynomial interpolation, it evaluates $C_{r,s}$ in Eq. (36).

$$C_{r,s} = P_1^{sr} P_2^{-sr} \hat{e}(t_u, C') \tag{36}$$

Finally, it gets back ciphertext partially decrypted in Eq. (37).

$$M' = C_{r,s} \hat{e}(t_u, C')^{-1} = P_1^{sr} P_2^{-sr} \tag{37}$$

CSP.DecGen$(params, M', CT_{\mathcal{T}}, s)$: It first evaluates Eq. (38) and returns Eq. (39).

$$k = H\left(M'^{s^{-1}} C''\right) \quad (38)$$

And then returns

$$M = Dec\,(k, C) \quad (39)$$

where M' - Partially decrypted ciphertext
tk_u - Decryption token
Att_u - Attribute Set
M - Message

Finally, the public parameter pk_{i_0} and secret parameter s_{i_0} are updated by the DA. Using the leafnode connected to i_0 in their access trees, the CSP re-encrypts ciphertexts by the new parameters. By the updated parameters, DA updates the secret keys for users of authenticated data connected to i_0.

4. Performance analysis and results

In this section, the proposed method ciphertext policy revocable and searchable attribute-based encryption (CP-RSABE) performance was evaluated by feature comparison to that of a few related works such as online/offline multi authority-ABE with cryptographic reverse firewalls (OO-MA-ABE-CRF) [15], ciphertext policy- attribute-based encryption (CP-ABE) [16], and attribute-based encryption with full privacy protection (ABE-FPP) [17]. The following performances like running time, communication overhead, and cost of storage are compared with the above-stated existing methods. Three expensive computing operations: the operation of pairing, and the exponential operations of G1, and G2 are used to calculate the execution time in asymptotic analysis. The existing methods were implemented with the same technique suitable to the research environment of the simulation defined Ubuntu 18.04 with an Intel core i5-2410M processor running at 2.3 GHz, 6 GB of RAM, and the cryptography of Java language is used to measure the real execution time and cost of storage. Also, consider the hash function as the SHA-1 algorithm, the XOR method as the encryption of the symmetric-key technique, and the and-gate access structures as the access control rules.

4.1 Characteristics comparison

The proposed system offers a mechanism of user revocation and flexible key delegation. In fact, in other multi-authority method, either an authority of single key supports the system in each data user or an authority of single key supports each attribute in the universe. However, data users and domain authorities are not subject to any restrictions. Additionally, the proposed system offers a mechanism of simple encryption and provides a quick decryption technique.

Table 1. Encryption time on the data owner side

Number of leaf nodes	OO-MA-ABE-CRF [15]	CP-ABE [16]	ABE-FPP [17]	Proposed method CP-RSABE
10	0.005	0.29	0.19	0.0112
20	0.009	0.35	0.26	0.0115
30	0.014	0.5	0.29	0.0119
40	0.017	0.65	0.35	0.0125
50	0.021	0.74	0.45	0.0131
60	0.025	0.8	0.64	0.0135
70	0.029	0.85	0.7	0.0142
80	0.032	0.9	0.78	0.0151
90	0.037	0.97	0.8	0.0156
100	0.041	1	0.85	0.0163

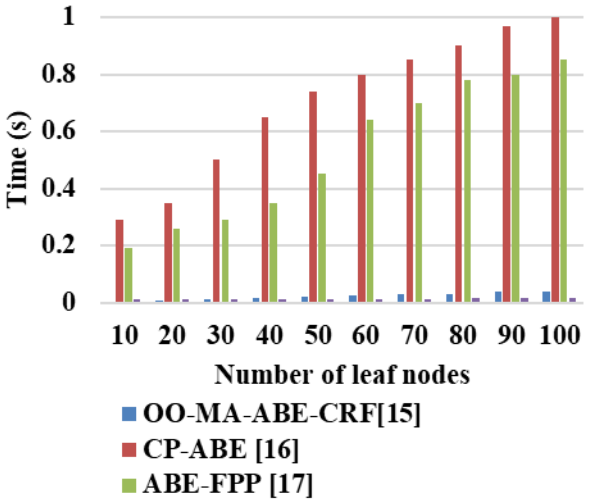

Figure. 2 Encryption time on the data owner side

Table 1. shows the encryption time of the proposed method CP-RSABE on different leaf nodes, respectively. The number of leaf nodes ranges from 10 to 100, which is typical of most ABE methods. After analyzing Fig. 2. it is noted that the result of proposed method has taken less time for encryption. IoT devices frequently need to perform decryption operations to decrypt control messages, whereas encrypted sensory information from the devices of IoT to clouds frequently has a higher priority. Table 1 illustrates that the proposed method CP-RSABE encryption method is significantly less and more efficient than OO-MA-ABE-CRF [15], CP-ABE [16], and ABE-FPP [17].

Furthermore, a comparison between the decryption time between CP-RSABE and existing methods OO-MA-ABE-CRF [15], CP-ABE [16], and

Table 2. Decryption time on the data user side

Number of Attributes	OO-MA-ABE-CRF [15]	CP-ABE [16]	ABE-FPP [17]	Proposed method CP-RSABE
10	0.04	0.03	0.03	0.02
20	0.08	0.06	0.06	0.05
30	0.12	0.08	0.08	0.06
40	0.14	0.12	0.12	0.10
50	0.18	0.14	0.14	0.12
60	0.22	0.16	0.16	0.14
70	0.24	0.18	0.18	0.16
80	0.28	0.23	0.23	0.20
90	0.31	0.24	0.24	0.22
100	0.34	0.27	0.27	0.25

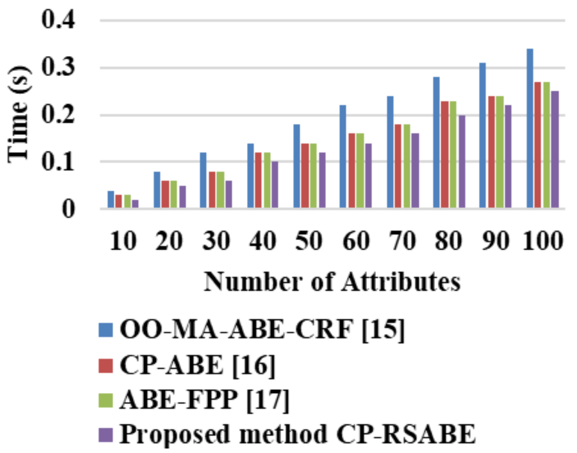

Figure. 3 Decryption time on the data user side

ABE-FPP [17] is shown in Fig. 3, decryption time is nearly identical to that of the existing method, and offer simple decryption methods. The number of attributes varies from 10 to 100 and the size varies from 0 to 5 are shown in Table 2. The decryption time of proposed method is more efficient than the other methods. Overall, the proposed CP-RSABE achieves improved computation performance in encryption and decryption methods and is more effective for resource-constrained devices while maintaining good security and user privacy when compared to state-of-the-art techniques.

Additionally, as shown in Fig. 4, proposed method reduces the communication cost. In particular, the cost of communication between the user and the authority or cloud sever depends less on how much attributes the user holds. Because while utilizing proposed method, the data owner only sends partial ciphertexts of lightweight, not entire ciphertexts to the CSP. The amount of overhead varies according to the size of the message being transferred between the entities are shown in Table 3. ABE generally includes the public keys, ciphertext, and private keys in this message. As a result, the size of the ciphertext, private and public keys are used to evaluate the communication overhead of the proposed method with existing methods, OO-MA-ABE-CRF [15], CP-ABE [16], and ABE-FPP [17]. However, it reduces the communication cost and performs significantly efficient than the existing schemes. Because while utilising CP-RSABE, the data owner only sends partial ciphertexts of lightweight, not entire ciphertexts, to the CSP.

Table 3. Communication overhead from the owner of data to the CSP

No of leaf nodes	OO-MA-ABE-CRF [15]	CP-ABE [16]	ABE-FPP [17]	Proposed method CP-RSABE
10	0.5	3	1	0.3
20	1	5.2	3	0.6
30	1.5	8	4	1.0
40	2	11	5	1.4
50	2.2	13	6	1.7
60	2.5	16	8	2.0
70	3	18	9	2.5
80	3.2	21	10	2.8
90	3.5	24	12	3.2
100	3.6	26	14	3.4

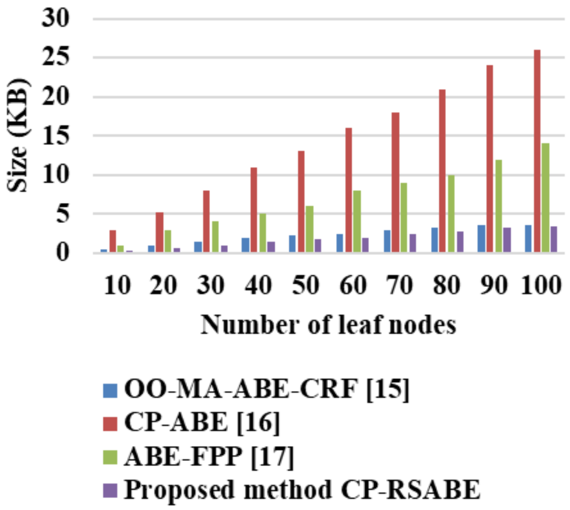

Figure. 4 Communication overhead from the owner of data to the CSP

4.2 Asymptotic and experimental results

The number of attributes varies from 10 to 100 and the proposed method performs low secret key size when compared to existing methods OO-MA-ABE-CRF [15], CP-ABE [16], and ABE-FPP [17] which is tabulated in Table 4. In contrast, the

Table 4. size of data user’s secret-key

Number of Attributes	OO-MA-ABE-CRF [15]	CP-ABE [16]	ABE-FPP [17]	Proposed method CP-RSABE
10	0.6	0.8	0.8	0.4
20	1.3	1.5	1.5	1.0
30	1.9	2.1	2.1	1.5
40	2.5	2.7	2.7	2.0
50	3.2	3.4	3.4	2.6
60	3.8	4	4	3.0
70	4.5	4.7	4.7	3.4
80	5.1	5.3	5.3	3.9
90	5.7	5.9	5.9	4.5
100	6.3	6.5	6.5	5.1

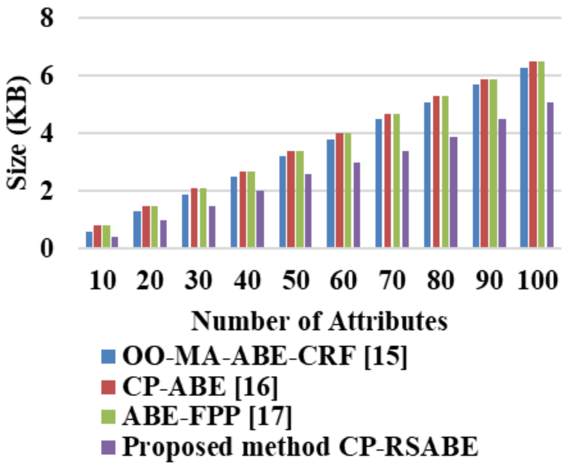

Figure. 5 Size of data user’s secret-key

proposed system associates each ciphertext with a policy of access control specified by a data owner and each data user’s secret-key with a set of attributes. A data user only recovers the data that is encoded in a ciphertext if the set of attributes representing the ciphertext complies with the access control policy of data users. Moreover, the proposed method links each ciphertext to a predetermined control policy of access provided by an owner of data while also linking a data user’s secret key to an attribute set. Fig. 5 shows the experimental results of the data user’s secret key. From the Fig. 5, it evidently shows that in comparison to other existing methods, the proposed system performs low secret key size of data user’s and it is efficient.

Table 5. Size of a ciphertext

Number of leaf nodes	OO-MA-ABE-CRF [15]	CP-ABE [16]	ABE-FPP [17]	Proposed method CP-RSABE
10	1.5	3	1.4	1.2
20	2.8	5.2	2.6	2.3
30	4	8.5	3.9	3.5
40	5.2	10.2	5.1	4.7
50	7	13.5	6.9	6.2
60	8.4	15.2	8.3	7.8
70	8.7	15.4	8.6	8.2
80	9.2	20.2	9	8.9
90	10.2	23.5	10.1	9.5
100	12.1	26.2	11.9	10.7

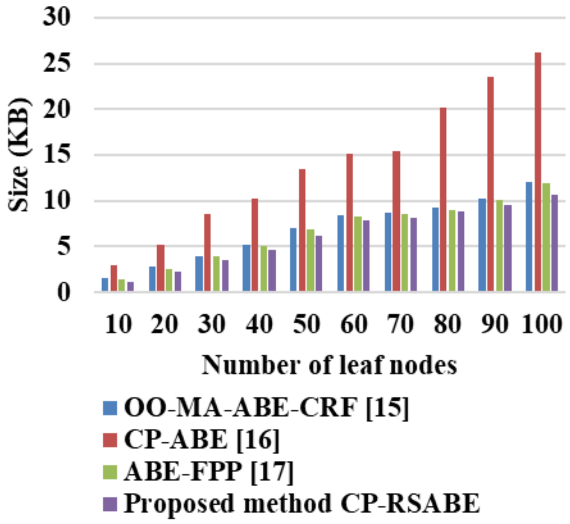

Figure. 6 Size of a ciphertext

Table 5 shows the ciphertext size of the proposed method CP-RSABE on different leaf nodes with existing methods OO-MA-ABE-CRF [15], CP-ABE [16], and ABE-FPP [17]. When the set of client characteristics complies with the policy of access and the search terms and indexes match, the cloud server returns the ciphertext search results. Ciphertext policy is protected by the CP-RSABE method. Each ciphertext is labelled by descriptive features set selected by an owner of data, and a data user’s secret key is connected to a control policy of access refused by the central authority. When the owner of data wants to share the data, it encrypts that data with the help of parameters to produce the ciphertext. A comparison between the size of a ciphertext between CP-RSABE and other existing methods is shown in Fig. 6. Moreover, it concludes that the proposed method storage overhead is acceptable.

4.3 Comparative analysis

The comparative analysis includes methods, encryption time, decryption time, communication overhead, size of data user’s secret key, and size of ciphertext. For better comparison, following methods OO-MA-ABE-CRF [15], CP-ABE [16], ABE-FPP

Table 6. Comparative analysis with the existing methods

Author	Method	Encryption time (s)	Decryption time (s)	Communication overhead (KB)	Size of data user's secret key (KB)	Size of ciphertext (KB)
Juyan Li (2023) [15]	OO-MA-ABE-CRF	0.041	0.34	3.6	6.3	12.1
Sowjanya (2021) [16]	CP-ABE	0.1	0.27	26	6.5	26.2
Tian (2021) [17]	ABE-FPP	0.85	0.27	14	6.5	11.9
Proposed method	CP-RSABE	0.0163	0.25	3.4	5.1	10.7

[17] are implemented and analyzed under same scenario. Table 6. shows that the comparative analysis of proposed method with the existing methods such as OO-MA-ABE-CRF [15], CP-ABE [16], and ABE-FPP [17].

While taking 100 leafnodes the existing methods such as OO-MA-CP-ABE method [15] has a (0.041s) encryption time, (0.34s) decryption time, (3.6KB) communication overhead, (6.3KB) size of data user's secret key, and (12.1KB) size of ciphertext. CP-ABE [16] has a (0.1s) encryption time, (0.27s) decryption time, (26KB) communication overhead, (6.5KB) size of data user's secret key, and (26.2KB) Size of ciphertext and so on. When compared with the existing methods CP-RSABE achieves the encryption time (0.0163s), decryption time (0.25s), communication overhead (3.4KB), size of secret key (5.1KB), and size of ciphertext (10.7KB).

4.4 Discussion

This section provides the discussion about the proposed CP-RSABE method and compared those results with existing methods OO-MA-CP-ABE [15], CP-ABE [16], and ABE-FPP [17] in comparative analysis 4.3. The major goal of this study is to protect privacy and security. The proposed method greatly lowers the cost of computing IoT devices with the availability of multiple-keyword searchers for the users of data. Additionally, the proposed system offers a mechanism of simple encryption and provides a quick decryption technique. The proposed method also provides user revocation and flexible, and scalable key delegation. The phases of key delegation and user revocation are individually handled by each of the key generator authorities and users are free to request their secret keys from whatever they choose the key authority. Additionally, the primary authority adds additional key generator authorities whether the system requires more computational power. In the result analysis, the CP-RSABE analyzed on different leaf nodes with a range from 10 to 100. When compared with three existing methods, CP-RSABE works better at all the leaf counts. While taking 100 leaf nodes, CP-RSABE significantly reduces the encryption time (0.0163s), decryption time (0.25s), communication overhead (3.4KB), size of secret key (5.1KB), and size of ciphertext (10.7KB).

5 Conclusion

In this paper, the ciphertext policy-revocable and searchable attribute-based encryption (CP-RSABE) method is proposed to protect privacy and security. Due to inefficiency and the absence of a straightforward attribute revocation mechanism with ABE methods, CP-RSABE provides advanced efficient outcomes. The proposed method also provides user revocation, flexible, and scalable key delegation. The performance and analysis of security showed that the proposed method is effective, secure, and appropriate for suitable applications of IoT. The primary authority adds additional key generator authorities whether the system requires more computational power. The CP-RSABE method performs significantly better in terms of ciphertext size, decryption time, and parameter size. While taking 100 leaf nodes, CP-RSABE significantly reduces the encryption time (0.0163s), decryption time (0.25s), communication overhead (3.4KB), size of secret key (5.1KB), and size of ciphertext (10.7KB). The proposed method can be more effective than the existing methods such as the OO-MA-CP-ABE, CP-ABE, and ABE-FPP. In the future, planning to add dynamic attribute revocation functionality in user revocation to improve the proposed CP-RSABE method.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

For this research work all authors' have equally contributed in conceptualization, methodology, validation, resources, writing—original draft preparation, writing—review and editing.

Notation description

Symbol	Description
$params$	Global public parameter
MSK	CA Master secret-key
U	Universal attribute key
λ	Security parameter
H	Hash function
π_S	Symmetric key
sk, pk	secret key, public key
sk_o	data owner secret key
pk_0	data owner public key
id_j	Identifier of the j-th DA
$\mathcal{T}$	Access tree
$CT_{\mathcal{T}}$	Ciphertext related with an access tree $\mathcal{T}$
$PCT_{\mathcal{T}}$	partial ciphertext related with an access tree $\mathcal{T}$
s_{i_0}	secret parameter
$\widetilde{pk_{\iota_0}}$	Updated public parameter associated with an attribute i_0
$UKey_{i_0}$	Update key associated with an attribute i_0
i_0	Attribute
$\widetilde{CT_{\mathcal{T}}}$	Re-encrypted ciphertext
M'	Partially decrypted ciphertext
tk_u	Decryption token
Att_u	Attribute Set
U_j	Attribute set supported by the j-th DA
PK_j	Public key of the j-th DA
MSK_j	Master Secret Key of the j-th DA
id_u	Identifier of the data user
M	Message
$\widetilde{s_{\iota_0}}$	Updated secret parameter associated with an attribute i_0
p_{i_0}	Public parameter

References

[1] A. K. Junejo, N. Komninos, and J. A. M. Cann, "A Secure Integrated Framework for Fog-Assisted Internet-of-Things Systems", *IEEE Internet of Things Journal*, Vol. 8, No. 8, pp. 6840-6852, 2021.

[2] S. Y. Tan, K. W. Yeow, and S. O. Hwang, "Enhancement of a Lightweight Attribute-Based Encryption Scheme for the Internet of Things", *IEEE Internet of Things Journal*, Vol. 6, No. 4, pp. 6384-6395, 2019.

[3] A. Diro, H. Reda, N. Chilamkurti, A. Mahmood, N. Zaman, and Y. Nam, "Lightweight Authenticated-Encryption Scheme for Internet of Things Based on Publish-Subscribe Communication", *IEEE Access*, Vol. 8, pp. 60539-60551, 2020.

[4] P. Chinnasamy, A. Albakri, M. Khan, A. A. Raja, A. Kiran, and J. C. Babu, "Smart Contract-Enabled Secure Sharing of Health Data for a Mobile Cloud-Based E-Health System", *Applied Sciences*, Vol. 13, No. 6, p. 3970, 2023.

[5] L. Zhang, W. You, and Y. Mu, "Secure Outsourced Attribute-Based Sharing Framework for Lightweight Devices in Smart Health Systems", *IEEE Transactions on Services Computing*, Vol. 15, No. 5, pp. 3019-3030, 2022.

[6] S. Banerjee, S. Roy, V. Odelu, A. K. Das, S. Chattopadhyay, J. J. P. C. Rodrigues, and Y. Park, "Multi-authority CP-ABE-based user access control scheme with constant-size key and ciphertext for IoT deployment", *Journal of Information Security and Applications*, Vol. 53, p. 102503, 2020.

[7] H. Kurdi and V. Thayananthan, "A Multi-Tier MQTT architecture with multiple brokers based on fog computing for securing industrial IoT", *Applied Sciences*, Vol. 12, No. 14, p. 7173, 2022.

[8] S. Sciancalepore, "PARFAIT: Privacy-preserving, secure, and low-delay service access in fog-enabled IoT ecosystems", *Computer Networks*, Vol. 206, p. 108799, 2022.

[9] M. Rasori, P. Perazzo, and G. Dini, "A lightweight and scalable attribute-based encryption system for smart cities", *Computer Communications*, Vol. 149, pp. 78-89, 2020.

[10] Z. Guan, W. Yang, L. Zhu, L. Wu, and R. Wang, "Achieving adaptively secure data access control with privacy protection for lightweight IoT devices", *Science China Information Sciences*, Vol. 64, p. 162301, 2021.

[11] S. Xu, Y. Li, R. H. Deng, Y. Zhang, X. Luo, and X. Liu, "Lightweight and expressive fine-grained access control for healthcare Internet-of-Things", *IEEE Transactions on Cloud Computing*, Vol. 10, No. 1, pp. 474-490, 2022.

[12] J. Sun, H. Xiong, X. Liu, Y. Zhang, X. Nie, and R. H. Deng, "Lightweight and Privacy-Aware Fine-Grained Access Control for IoT-Oriented Smart Health", *IEEE Internet of Things Journal*, Vol. 7, No. 7, pp. 6566-6575, 2020.

[13] J. Hao, C. Huang, J. Ni, H. Rong, M. Xian, and X. S. Shen, "Fine-grained data access control with attribute-hiding policy for cloud-based IoT", *Computer Networks*, Vol. 153, pp. 1-10, 2019.

[14] K. Zhang, J. Long, X. Wang, H. N. Dai, K. Liang, and M. Imran, “Lightweight Searchable Encryption Protocol for Industrial Internet of Things”, *IEEE Transactions on Industrial Informatics*, Vol. 17, No. 6, pp. 4248-4259, 2021.

[15] J. Li, Y. Fan, X. Bian, and Q. Yuan, “Online/Offline MA-CP-ABE with Cryptographic Reverse Firewalls for IoT”, *Entropy*, Vol. 25, No. 4, p. 616, 2023.

[16] 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, p. 102108, 2021.

[17] H. Tian, X. Li, H. Quan, C. C. Chang, and T. Baker, “A Lightweight Attribute-Based Access Control Scheme for Intelligent Transportation System With Full Privacy Protection”, *IEEE Sensors Journal*, Vol. 21, No. 14, pp. 15793-15806, 2021.

[18] A. Saidi, O. Nouali, and A. Amira, “SHARE-ABE: an efficient and secure data sharing framework based on ciphertext-policy attribute-based encryption and Fog computing”, *Cluster Computing*, Vol. 25, No. 1, pp. 167-185, 2022.

[19] S. F. Aghili, M. Sedaghat, D. Singelée, and M. Gupta, “MLS-ABAC: efficient multi-level security attribute-based access control scheme”, *Future Generation Computer Systems*, Vol. 131, pp. 75-90, 2022.

[20] S. Fugkeaw, L. Wirz, and L. Hak, “Secure and Lightweight Blockchain-enabled Access Control for Fog-Assisted IoT Cloud based Electronic Medical Records Sharing”, *IEEE Access*, Vol. 11, pp. 62998-63012, 2023.