Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.041167

ARTICLE

# Efficient Multi-Authority Attribute-Based Searchable Encryption Scheme with Blockchain Assistance for Cloud-Edge Coordination

**Peng Liu[1], Qian He[1,*], Baokang Zhao[2], Biao Guo[1] and Zhongyi Zhai[1]**

[1]School of Computer and Information Security, Guilin University of Electronic Technology, Guilin, 541004, China

[2]School of Computer Science, National University of Defense Technology, Changsha, 410073, China

*Corresponding Author: Qian He. Email: heqian@guet.edu.cn

Received: 13 April 2023 Accepted: 26 June 2023 Published: 08 October 2023

**ABSTRACT**

Cloud storage and edge computing are utilized to address the storage and computational challenges arising from the exponential data growth in IoT. However, data privacy is potentially risky when data is outsourced to cloud servers or edge services. While data encryption ensures data confidentiality, it can impede data sharing and retrieval. Attribute-based searchable encryption (ABSE) is proposed as an effective technique for enhancing data security and privacy. Nevertheless, ABSE has its limitations, such as single attribute authorization failure, privacy leakage during the search process, and high decryption overhead. This paper presents a novel approach called the blockchain-assisted efficient multi-authority attribute-based searchable encryption scheme (BEM-ABSE) for cloud-edge collaboration scenarios to address these issues. BEM-ABSE leverages a consortium blockchain to replace the central authentication center for global public parameter management. It incorporates smart contracts to facilitate reliable and fair ciphertext keyword search and decryption result verification. To minimize the computing burden on resource-constrained devices, BEM-ABSE adopts an online/offline hybrid mechanism during the encryption process and a verifiable edge-assisted decryption mechanism. This ensures both low computation cost and reliable ciphertext. Security analysis conducted under the random oracle model demonstrates that BEM-ABSE is resistant to indistinguishable chosen keyword attacks (IND-CKA) and indistinguishable chosen plaintext attacks (IND-CPA). Theoretical analysis and simulation results confirm that BEM-ABSE significantly improves computational efficiency compared to existing solutions.

**KEYWORDS**

Attribute-based encryption; search encryption; blockchain; multi-authority; cloud-edge

## 1 Introduction

The widespread use of the Internet of Things (IoT) and 5G have led to a surge in the number of network edge devices, resulting in a rapid growth in edge data [1,2]. The centralized data processing approach based on cloud computing is facing challenges in efficiently processing the vast amount of data generated by edge devices. Edge computing has emerged as a promising solution to the challenges faced by traditional cloud computing in processing the massive amounts of data generated by IoT devices. The fundamental concept of edge computing is to perform computing tasks close to the data

source, which reduces network transmission bandwidth and response delay compared to traditional cloud computing [3]. However, the untrusted or partially trusted nature of cloud service providers (CSP) and edge nodes (ENs) poses a significant risk to the privacy of sensitive data. Tampering and abusing data by these entities can leak user privacy information [3–5]. Although symmetric encryption can be used by the data owner (DO) to maintain data confidentiality, the use of encryption prevents the ability to perform plaintext keyword retrieval. It creates challenges for fine-grained access control and data sharing.

To mitigate the potential risks of private data leakage, it is crucial to prioritize both data confidentiality and accessibility for effective access control. While symmetric encryption can provide data confidentiality, it makes information on encrypted data difficult to retrieve. Identity-based encryption (IBE) and attribute-based encryption (ABE) provide distinct access control mechanisms, with IBE offering coarse-grained access control and ABE providing fine-grained access control capabilities [6,7]. It is critical in practice to have an effective keyword search and to have fine-grained access control over encrypted data. The technique of searchable encryption (SE) enables data users (DUs) to conduct searches on ciphertext data using specific keywords [8]. To provide even more precise access control, the gradually popular solution in both industrial and academic domains is ciphertext-policy attribute-based searchable encryption (CP-ABSE) with flexible access control policies [9,10]. The high computational and storage requirements of CP-ABSE prevent its deployment on resources-constrained IoT devices, despite its promise as a SE scheme for fine-grained access control. Therefore, the lightweight CP-ABSE scheme is a prerequisite for its implementation on resource-constrained terminal devices. Additionally, many existing CP-ABSE schemes [9–13] that employ single-attribute authorization for attribute management and key distribution may encounter challenges in efficiently and securely handling attributes from a vast network of interconnected IoT devices and are prone to single-point failures and central corruption. Furthermore, trust in CSP is often weakened due to the risk of malicious access to data and tampering with query results. In contrast, blockchain technology provides a safer and more trustworthy option [14]. As a decentralized ledger with multi-party consensus and a chain structure, blockchain offers an unparalleled guarantee of data integrity and trustworthiness compared to centralized systems.

This paper proposes an efficient multi-authority attribute-based searchable encryption scheme with blockchain assistance (BEM-ABSE) for cloud-edge collaboration. This BEM-ABSE scheme aims to provide secure and reliable searching while protecting privacy through blockchain, ciphertext searching, and ABE. To address the efficiency limitations and security vulnerabilities associated with Certificate Authorities (CAs), the BEM-ABSE scheme employs a consortium blockchain, enabling multiple attribute authorities to autonomously manage user attributes and key assignments. Furthermore, this scheme facilitates online/offline hybrid encryption and edge-assisted verifiable decryption, effectively minimizing the computational overhead involved in encryption and decryption operations. The main contributions of the scheme are as follows:

(1) Taking advantage of multi-authority ABE and blockchain, this paper proposes a searchable encryption scheme with fine-grained access control for cloud-edge collaboration. BEM-ABSE supports ciphertext keyword search based on smart contracts, online/offline hybrid encryption, and edge-assisted verifiable outsourcing decryption. This paper also proves this scheme can resist IND-CPA and IND-CKA under the random oracle model.

(2) Consortium blockchain is designed to replace the trusted CA in the traditional CP-ABSE scheme, allowing for the generation of global public parameters and the execution of ciphertext

searches. The dependence on single-centre authorization is broken, and the reliability of ciphertext searches is improved.

(3) An online/offline hybrid encryption mechanism is utilized to reduce the time overhead during the encryption phase by performing pre-encryption computation and generating intermediate ciphertext. The decryption tasks are offloaded to ENs, effectively decreasing the computational burden of decryption for resource-constrained IoT devices.

The remainder of this paper is organized as follows. Section 2 provides a review of the related work. Section 3 demonstrates the background knowledge in the understanding of the BEM-ABSE. In Section 4, Formalize the system and security model. Then, the formal construction of the BEM-ABSE scheme is presented in Section 5. In Section 6 and 7, separate analyses of safety and performance are presented. Finally, The work of this paper is concluded in Section 8.

## 2 Related Work

### *2.1 Search Encryption*

SE enables search on encrypted data using specified keywords, while ABSE provides detailed permissions control for data ciphertext retrieval, with significant research having been conducted in this field. Searchable symmetric encryption was first proposed by Song et al. [8] in 2000. However, using a single shared key for encryption and decryption in symmetric cryptography makes it impractical for complex multi-user applications. ABSE provides a flexible way to execute access control policies, ensuring that only users with the required policy attributes can access data. This one-to-many access control model enables secure and convenient data sharing. To reduce the computational overhead during the search process, Zheng et al. [9] proposed an ABSE scheme with verifiable results, which uses verifiable attribute-based encryption, but it also has some drawbacks, such as requiring a secure channel and high costs. Huang et al. [15] introduced a rapid and privacy-preserving attribute-based keyword search system designed for cloud document services. This system exhibits improved stability and efficiency during the search phase, but it does entail additional computational costs in other phases. Zhang et al. [16] designed a distributed and scalable, searchable encryption access control scheme that utilizes cloud services to achieve lightweight decryption processes, resulting in lower computational complexity and improving security against selected keyword attacks and selected plaintext attacks, but not suitable for resource-constrained devices due to high encryption time overhead. Considering the limitations of resource-constrained devices, Miao et al. [17] proposed a constant-sized trapdoor-based online/offline SE for cloud-assisted industrial IoT, where the overall encryption burden on DO is still heavy, but the cost of generating DU's trapdoor is reduced through an elegant technique. Zhou et al. [18] proposed a general searchable encryption scheme for cloud-assisted industrial IoT systems, with the lightweight generation of both index and query trapdoors. Liu et al. [5] proposed an efficient ABSE scheme for cloud-edge collaborative computing, reducing the computational cost of resource-constrained terminals by allowing EN to simultaneously perform text-based search and pre-decryption algorithms and save keyword indexes.

However, these schemes risk privacy data leakage as the CSP and ENs are either untrusted or semi-trusted. The combination of searchable public key encryption with blockchain technology is gaining popularity among scholars to enhance ciphertext security. This approach benefits from blockchain technology's decentralized, transparent, traceable and tamper-proof characteristics. Yang et al. [19] presented a scheme allowing encrypted file upload to the cloud while placing the encrypted index on the blockchain. This scheme ensures the encrypted index is tamper-proof, integrity, and traceability and enables users to obtain accurate search results without needing third-party verification. However,

these schemes have limitations, such as scalability difficulties, security and performance bottlenecks, and the potential for excessive permissions, as they rely on a single authorization center. Niu et al. [20] proposed a policy hide and verifiable blockchain-assisted ABSE scheme. This scheme stores the index is stored on the blockchain, and searches are performed using smart contracts, which reduces the computational load on the service. With the growth of the Internet of Things and the widespread adoption of 5G wireless networks, the cloud-edge collaborative data-sharing model has become more prevalent, and the number of IoT devices requiring authorization has increased significantly. However, relying on a single authorization center can result in significant losses if it crashes or is compromised.

### *2.2 Multi-Authority Attribute-Based Encryption*

There are significant security risks in the current ABE schemes, as they rely on one attribute authority to manage attributes and keys. This authority may be able to decrypt any ciphertext within its control. To address this issue, researchers have proposed a variety of multi-authority ABE schemes (MA-ABE). The MA-ABE scheme was first proposed by Chase [21], but managing attribute authorities requires a trusted certificate authority, which may prove costly and have backward security challenges. Subsequently, Lewko et al. [22] proposed a distributed multi-authority ABE where attribute authorities are solely responsible for creating initial public parameters. The scheme utilizes a linear secret sharing scheme (LSSS) matrix to represent access policies, offering greater expressive capabilities compared to AND gates. However, the scheme lacks post-quantum security assurance. Tu et al. [23] suggested using attribute-group keys for large attribute domains in distributed computing systems using fog computing. To improve user privacy and security, Guo et al. [24] developed an encrypted data access control solution that utilizes smart contracts to define interactions between DOs, users and attribute authorities. However, DOs using symmetric encryption for data encryption can lead to heavy key management overhead. Qin et al. [25] utilized a consortium blockchain to establish trust bridges between attribute authorities and designed an MA-ABE based on blockchain. However, the existence of certificate authorities raises concerns about potential single-point failures. According to Xiao et al. [26], their blockchain-based MA-ABE scheme incorporates flexible attribute revocation; It can be applied to data publishing services and payment platforms for Dos. To manage dynamic users and improve search result credibility, Yu et al. [27] proposed an efficient multi-authority SE scheme using blockchain technology for keyword-based search and dynamic user management. Multiple-cloud block storage technology was used by Wu et al. [28] to address the problems with unstable cloud servers and to guard against malicious actions, including the leakage of private information, tampering with ciphertext, and malicious deletion of ciphertext. The security of keyword indexes and the impartiality of search results are guaranteed by the blockchain's immutability. Utilizing online/offline encryption and outsourced decryption processes, Xu et al. [29] distributed ABSE approach with shared keyword search was suggested. Although the key delegation problem is resolved inside a single authority, the approach has a somewhat high total computation cost.

While these schemes address the attribute and key management issue a single authority brings, current multi-authority systems still face some challenges. Some schemes rely on a central authority for management [30], generating complete private keys through the CA to avoid single-point of failure of attribute authorities. However, this approach also involves high trust costs for the CA. Additionally, there is over-reliance on the cloud service when users send encryption requests to the cloud. The CSP usually performs encryption search and pre-decryption processes [31], meaning they can arbitrarily modify the search results or encryption data.

## 3 Preliminaries

### 3.1 Bilinear Maps

Assume that $G$ and $G_T$ are multiplicative cyclic groups, where the group order is $p$, and the generator is $g$. The properties described below apply to the bilinear group mapping: $e: G \times G \rightarrow G_T$:

(1) *Biplanarity*: $e\left(u^a, v^b\right) = e(u, v)^{ab}$, $\forall u, v \in G$ and $a, b \in Z_p$;
(2) *Nondegeneracy*: $e(g, g) \neq 1$;
(3) *Computability*: an efficient algorithm exists to calculate $e(u, v)$, $\forall u, v \in G$;

### 3.2 Access Structure

***Definition* 1 (*access structure*):** Given that there are $n$ participants $\mathcal{P} = \{P_1, P_2, \cdots, P_n\}$. A collection $\mathbb{A} \subseteq 2^{\{P_1, P_2, \cdots, P_n\}}$ considered monotonic when the following conditions are satisfied: $\forall B, C$: if $C \in \mathbb{A}$ holds on condition that $B \subseteq C$. A monotonic access structure is defined as a collection $\mathbb{A}$ containing non-empty subsets $\{P_1, P_2, \cdots, P_n\}$. Authorized collections are those within the collection $\mathbb{A}$, while unauthorized collections refer to the remaining subsets.

### 3.3 Pedersen (t, n) Secret Sharing Algorithm

Each participant is both a distributor and a participant in the Pedersen ($t$, $n$) [27]. Given that there are $n$ participants $p = (p_1, p_2, \cdots, p_n)$ and distribute the respective sub-secret $S_i$ using the Shamir secret-sharing algorithm. The specific design of the algorithm is outlined as follows.

(1) Generating the master secret $S$: Each distributor (participant) $p_i$ randomly selects their respective sub-secret $S_i$. The master secret of the whole scheme is defined as $S = \sum_{i=1}^{n} S_i$.

(2) Producing the sub-share value $s_{i,j}$: $p_i$ chooses a $t$-1th degree polynomial $f_i(x)$ satisfying $S_i = f_i(0)$ and calculates $s_{i,j} = f_i\left(x_j\right)_{j \in [1,n]}$ for each $p_i$. Then, $p_i$ sends the sub-share $s_{i,j}$ to the associated participant $p_i$ and keeps $s_{i,i}$ as part of the main share.

(3) Producing the master share $s_i$: Each $p_i$ calculates the respective $s_i$ with the formula as $s_i = \sum_{j=1}^{n} s_{j,i}$, where $s_{i,j}$ is the share held by participant $p_i$ itself. Note that $p_i$ just presents the master share $s_i$ when reconstructing the secret as a sub-share of the reconstructed secret.

(4) Recovering the master secret: If any $t$ participants can recover the master secret, it may be assumed that $p_1, p_2, \cdots, p_t$ have the capacity to rebuild the $S$ using the Lagrange interpolation formula and the specific algorithm is $S = \sum_{i=1}^{t} s_i \sum_{j=1, j \neq i}^{k} \frac{-j}{i-j}$.

The Pedersen ($t$, $n$) algorithm achieves secure sharing of secrets among multiple participants without revealing any information about the secret without a trust center. Therefore, it is executed by blockchain nodes in the BEM-ABSE scheme to produce global parameters and accomplish ciphertext search.

### 3.4 Security Assumptions

***Definition* 2 (*Bilinear Diffie-Hellman* (BDH)** assumption). Let $(G, G_T, g, e)$ as the bilinear mapping parameter and elements $g^a, g^b, g^c \in G$, where $a, b, c \in Z_p$ are selected random, The BDH problem in $(G, G_T, g, e)$ is hard to compute the bilinear pairing $e(g, g)^{abc} \in G_T$ from $g^a, g^b, g^c$. The algorithm has the advantage $\varepsilon$ in solving the BDH problem in the group $G$ when the following inequation Eq. (1) holds. The BDH assumption is true as long as the algorithm $\mathcal{A}$ is never able to solve the BDH issue

satisfactorily by a non-negligible margin.

$$\left|\Pr\left[\mathcal{A}\left(g, g^{a}, g^{b}, g^{c}\right) = e\left(g, g\right)^{abc}\right]\right| \geq \varepsilon \tag{1}$$

***Definition* 3 (*Decisional q-parallel Bilinear Diffie-Hellman Exponent* (BDHE)** assumption). Let $(G, G_T, g, e)$ as the bilinear mapping parameter and $a, s, b_1, \cdots, b_q \in Z_p$ as the random elements. Given:

$$y = \begin{bmatrix} g, g^{s}, g^{a}, \cdots, g^{a^{q}}, g^{a^{q+2}}, \cdots, g^{a^{2q}}, \\ \forall 1 \leq i \leq q g^{sb_i}, g^{a/b_i}, \cdots, g^{a^{q}/b_i}, g^{a^{q+2}/b_i}, \cdots, g^{a^{2q}/b_i}, \\ \forall 1 \leq i, j \leq q, i \neq j g^{asb_j}, \cdots, g^{a^{q}sb_j/b_i} \end{bmatrix} \tag{2}$$

Even though the adversary has a tuple $y$, the tuple $e\left(g, g\right)^{a^{q+1}s}$ and a random element $R \in G_T$ can nonetheless be difficult to differentiate from one another.

$$\left|\Pr\left[\mathcal{B}\left(y, T = e\left(g, g\right)^{a^{q+1}s}\right) = 0\right]\right| - \left|\Pr\left[\mathcal{B}\left(y, T = R\right) = 0\right]\right| \geq \varepsilon \tag{3}$$

When the inequality Eq. (3) is satisfied, the algorithm demonstrates an advantage E in solving the q-BDHE problem. This implies that it is not possible for any algorithm to successfully solve the decisional q-BDHE problem with non-negligible advantage.

## 4 Scheme Design

### *4.1 System Architecture*

The system architecture of the suggested strategy is displayed in Fig. 1. It comprises five entities: DO, EN, CSP, AAs, DU and BC. The BEM-ABSE scheme system model is depicted in Fig. 1, demonstrating the scheme's fundamental structure.

1) **DO.** Any IoT device capable of generating data. DO sets access policies, encrypts files and keyword indexes, and uploads the encrypted data and keyword indexes over a wireless network to EN.

2) **EN.** ENs are located at the edge of the network and possess strong computing and storage capabilities. They are able to dutifully store the ciphertext in the CSP and embed the keyword index and ciphertext address into the keyword index storage transaction, which is then submitted to the blockchain. In addition, ENs assist DU in partially decrypting the ciphertext, but they cannot obtain any information during the decryption process.

3) **CSP.** CSP is responsible for providing storage services for the encrypted data uploaded by legitimate DO through EN. In addition, it allows EN to access the ciphertext data associated with search results.

4) **AAs.** The BEM-ABSE has a number of attribute authorities. Each AA manages multiple attributes in an attribute domain and generates user attribute keys based on its user attributes.

5) **DU.** DUs create search trapdoors using keywords of their interest and embed them into search transactions, which are then submitted to the blockchain for subsequent encrypted file searching. After receiving partially decrypted ciphertext associated with search results from the EN, DUs can fully decrypt the data using their identity private keys.

6) **BC.** BC consists of trusted nodes responsible for global parameter generation and user registration. Search smart contracts (SSC) and validation smart contracts (VSC) are deployed on the blockchain. SSC conducts encrypted file searching on the blockchain through search trapdoors submitted by users, while VSC verifies the integrity of the data associated with user search results.

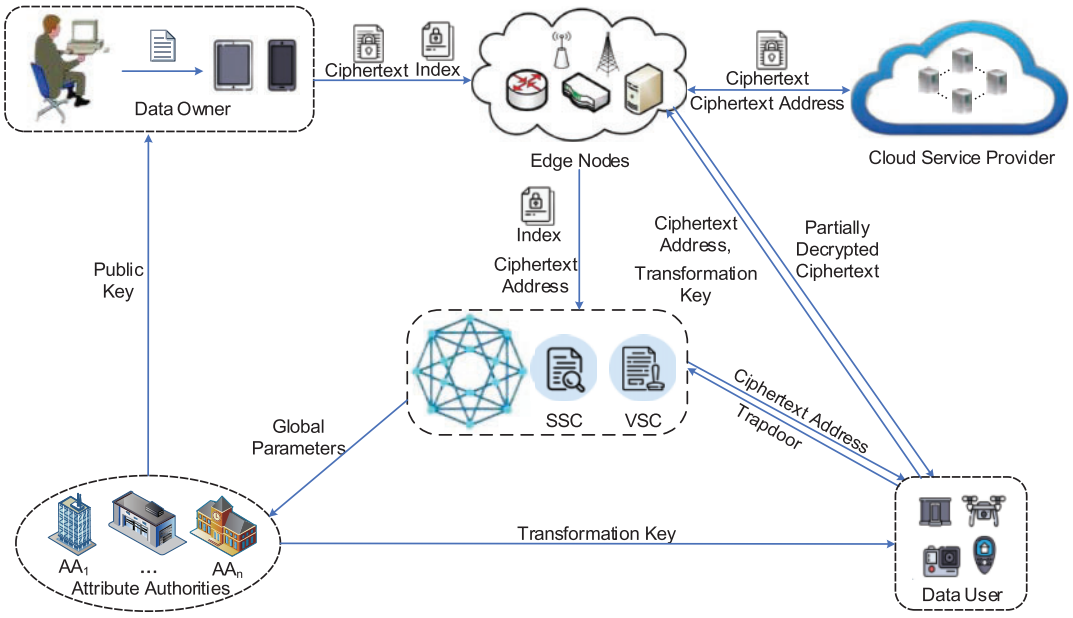


**Figure 1:** System architecture

EN serves as a crucial link between users and the cloud in the BEM-ABSE. DO encrypts and transmits a large amount of generated data to the cloud through EN, reducing the cost of local storage management. Moreover, in order to lessen the computing burden of the decryption process, the EN nearest to DU is in charge of partly decrypting the ciphertext. CSP is solely responsible for storing a large amount of encrypted data. A permission blockchain composed of pre-selected trusted nodes is accountable for storing encrypted indices, conducting ciphertext searches, and verifying decryption results to achieve secure and controllable encrypted retrieval.

### *4.2 Scheme Definition*

The BEM-ABSE scheme includes the following nine algorithms. Assuming there are $N$ attribute authorities $\{AA_1, AA_2, \cdots, AA_N\}$ in the BEM-ABSE, a global property set $S$ has a total of $U$ attributes, and each AA manages a different set of attributes $S_i, i \in N$.

(1) Setup

1) $GlobalSetup(1^{\lambda}) \rightarrow GP$. The BC executes the algorithm. Given a security parameter $\lambda$, and then outputs the global parameters $GP$.

2) $AuthoritySetup\ (GP, S_i) \rightarrow (PK_i, SK_i)$. Given the $GP$ and $S_i$, Each $AA_i$ runs the procedure to produce the public and private keys $(PK_i, SK_i)$. Notice that the $SK_i$ is held by attribute authority.

(2) Key Generation

1) $IdKeyGen\ (GP, uid) \rightarrow (usk_{uid}, upk_{uid})$. Given the user identification $uid$ and $GP$, the legitimate user conducts the algorithm to output its secret key $usk_{uid}$ and public $upk_{uid}$. Notice that the $usk_{uid}$ is held by the user and send $upk_{uid}$ to BC for registration.

2) $SKGen\,(GP, uid, S_{uid}, SK_i, PK_i) \rightarrow SK_{i,uid}$. Given the $GP$, $uid$, user attribute set $S_{uid}$, $PK_i$ and $SK_i$. Each associated $AA_i$ executes this algorithm to generate the decryption key $SK_{i,uid}$ and sends it to DU to construct the user transform key $TK_{uid}$.

(3) Encryption

1) $Offline.Enc\,(GP, PK_i) \rightarrow IC$. This phase is performed by the DO's more computationally capable devices. It takes the $GP$ and $PK_i$ as input and outputs intermediate ciphertext $IC$. Note that this part of the operation is calculated only once when the set of attributes of DO remains unchanged.

2) $Online.Enc\,(GP, IC, PK_i, (M_{l\times n}, \rho), F, \mathcal{W}) \rightarrow (CT, I_w)$. Given access policy $(M_{l\times n}, \rho)$, original data $F$, keyword set $\mathcal{W}$ and $GP$, $IC$, $PK_i$. It generates a set $I_w$ of keyword indexes and ciphertext $CT$.

(4) Trapdoor Generation

$TrapGen\,(GP, w') \rightarrow T_{w'}$. Given $GP$ and an interesting keyword $w'$. DU executes the algorithm to generate trapdoor $T_{w'}$ related to the $w'$.

(5) Search

$Search\,(T_{w'}, I_w) \rightarrow CT/\perp$. Taking $T_{w'}$ and $I_w$ as inputs, SSC runs a search algorithm to search for the file that matches the trapdoor $T_{w'}$ with the index $I_w$. Afterward, the address in ciphertext linked with the search results is sent to DU by SSC.

(6) Decryption

1) $EN.Dec\left(CT, GP, SK_{i,u}\right) \rightarrow CT'$. When receiving the $CT$ obtained from CSP using the ciphertext address from DU, take as input $GP$ and transformation key $TK_{uid}$ of the user $uid$, the EN generation the partial decrypt ciphertext $CT'$ for DU.

2) $User.Dec\,(CT', usk_{uid}, VK_F) \rightarrow F/\perp$. After gaining the $CT'$ from the EN, DU decrypts the $CT'$ using its $usk_{uid}$ to obtain the symmetric key, thus recovering the data file $C_F$.

### *4.3 Security Model*

The security of BEM-ABSE is based on the BDH assumption and $q$-BDHE assumption. This paper design two security games to demonstrate that the BEM-ABSE system is secure in the IND-CKA and IND-CPA models.

(1) IND-CKA mode.

The BEM-ABSE scheme is IND-CKA secure. A pre-selected group of reliable and secure nodes serves as the consensus node in a blockchain, albeit these nodes might be unavailable or infected. As long as the Pedersen ($t$, $n$) secret sharing method remains safe, no one node can independently complete the reconstruction of the system's secret parameters, keeping the entire blockchain secure. The IND-CKA of the BEM-ABSE is defined as a game between challenger $\mathcal{C}$ and adversary $\mathcal{A}$.

***Setup*:** The challenger $\mathcal{C}$ invokes the Pedersen algorithm to run the *GlobalSetup* generate $GP$ and sends the $GP$ to the $\mathcal{A}$.

***Phase* 1:** In polynomial-time many times (PPT), $\mathcal{A}$ provides a keyword collection to $\mathcal{C}$, then $\mathcal{C}$ performs *TrapGen* to generate trapdoor associated with each keyword and sends them to the adversary.

***Challenge*:** $\mathcal{A}$ provides challenge query keywords $w_0^*$ and $w_1^*$ that without appearing in *Phase* 1, where the lengths of keywords are the same. Then, $\mathcal{C}$ choose an arbitrary bit $b$ and runs *Online.Enc* algorithm to generate $I_{w_b}^*$ for the $w_b^*$, then $\mathcal{C}$ returns the $I_{w_b}^*$ to adversary.

***Phase* 2:** $\mathcal{A}$ adaptively repeats the execution of query *Phase* 1, while it should follow the constraints of the query phase.

***Guess*:** $\mathcal{A}$ outputs its guessed bit $b'$, and if $b = b'$, $\mathcal{A}$ wins the attack game; Otherwise, it fails. The advantage of $\mathcal{A}$ winning this game is $Adv_{\mathcal{A}}^{IND-CKA} = \left|\Pr[b = b'] - \frac{1}{2}\right|$.

***Definition* 4:** If the bilinear Diffie-Hellman assumption holds, the BEM-ABSE scheme achieves IND-CKA security.

(2) IND-CPA mode.

The BEM-ABSE scheme is IND-CPA secure. Assume that $\mathcal{A}$ can adaptively perform any key query while only statically corrupting the attribute authority. Let $S_{\mathrm{A}}$ be a set of AAs and $S$ be a set of attributes. The IND-CPA of the BEM-ABSE is defined in a game between $\mathcal{C}$ and $\mathcal{A}$.

***Init*:** The adversary $\mathcal{A}$ pre-selection of the corrupted set of attribute authorities is $S_{\mathrm{A}}^{'} \subseteq S_{\mathrm{A}}$ and chooses an $(M^*, \rho^*)$. After that, $\mathcal{A}$ provides this access structure to $\mathcal{C}$. In addition, the $\mathcal{A}$ construct and initialize collection $\mathbb{D}$ and table $\mathbb{T}$.

***Setup*:** $\mathcal{C}$ invokes Pedersen algorithm runs *GlobalSetup* generation *GP*, and sends the *GP* to $\mathcal{A}$. At the same time, the $\mathcal{C}$ performs *AuthoritySetup* on the attribute authority in the set $S_{\mathrm{A}} - S_{\mathrm{A}}^{'}$ to generate the key pair $(PK, SK)$ and returns $PK$ back to $\mathcal{A}$. For the attribute authority corrupted in the set $S_{\mathrm{A}}^{'}$, the $\mathcal{A}$ direct performs *AuthoritySetup* to obtain key pairs.

***Phase* 1:** $\mathcal{A}$ sends $(S, S_{uid}, uid)$ to challenger $\mathcal{C}$ for the decryption key $SK_{i,uid}$ query with the user attribute set $S_{uid}$, global set of attributes $S = \{S_i\}_{i \in S_{\mathrm{A}} - S_{\mathrm{A}}^{'}}$ and user identification $uid$. Notice that, there is a restriction that the set $S_{uid}$ of all attributes in $\mathbb{D}$ cannot satisfy $(M^*, \rho^*)$. If $\mathcal{A}$ has previously submitted $S_{uid}$, then $\mathcal{C}$ retrieves the entity $(S, S_{uid})$ by searching $\mathbb{T}$ and returns $SK_{S,uid}$. Otherwise, $\mathcal{C}$ performs *SKGen* algorithm to generate $SK_{S,uid} = \left\{S_{i,uid}\right\}_{i \in S_{\mathrm{A}} - S_{\mathrm{A}}^{'}}$ and sets $\mathbb{D} = \mathbb{D} \cup S_{uid}$.

***Challenge*:** $\mathcal{A}$ provides challenge cipertext $m_0^*$ and $m_1^*$, where the lengths of keywords are the same. Then, the $\mathcal{C}$ randomly selects bit $b$ and encrypts $m_b^*$ with $(M^*, \rho^*)$ and runs the *Online.Enc* algorithm to generate $CT_b^*$, Finally, $\mathcal{C}$ returns $CT_b^*$ back to adversary.

***Phase* 2:** $\mathcal{A}$ adaptively repeats the execution of query *Phase* 1, while it should follow the constraints of the query phase.

***Guess*:** $\mathcal{A}$ outputs its guessed bit $b'$, and if $b = b'$, $\mathcal{A}$ wins the attack game; Otherwise, it fails. The advantage of $\mathcal{A}$ winning this game is $Adv_{\mathcal{A}}^{IND-CKA} = \left|\Pr[b = b'] - \frac{1}{2}\right|$.

***Definition* 5:** The BEM-ABSE scheme achieves IND-CPA security if no PPT adversary has a significant advantage in the security game described above.

## 5 BEM-ABSE Construction

(1) Setup

1) *GlobalSetup*: This algorithm runs through the primary node with inputs of security parameter $\lambda$, the algorithm first creates the symmetric bilinear pairing $(G, G_T, p, g, e)$, where $G$ and $G_T$ are cyclic groups of the same prime order $p$ with a generator $g$. Next, the node shares four parameters $a, \mu, c, \gamma \in Z_p^*$ using the Pedersen secret sharing protocol. Blockchain node $BN$ calculates $g^{a_{BN}}, g^{\mu_{BN}}, g^{c_{BN}}, g^{\gamma_{BN}}$ based on their shared secret shares $a_{BN}, \mu_{BN}, c_{BN}, \gamma_{BN} \in Z_p^*$, respectively, and broadcast these values

to other nodes in the network. Then, the node uses three hash functions $H : \{0,1\}^* \rightarrow Z_p^*$, $H_1 : \{0,1\}^* \rightarrow G$ ,$H_2 : G_T \rightarrow \{0,1\}^{log_p}$ to simulate a random oracle model. Finally, the system publishes the global parameters with Eq. (4).

$$GP = (G, G_T, p, g, e, H, H_1, H_2, g^a, g^\mu, g^c, g^\gamma) \tag{4}$$

2) *AuthoritySetup*: Each attribute authority $AA_i$ randomly selects a element $\alpha_i \in Z_p^*$ and calculates $y_i = g^{\alpha_i}$. Then, $AA_i$ randomly selects $u_j \in Z_p^*$ for each of the managed attributes $a_j \in S_i$. Note that each attribute authority manages a unique attribute set. Finally, it keeps the private key secret $SK_i = \left(\alpha_i, u_j\right)$ and reveals its $PK_i = \left(e\left(g,g\right)^{\alpha_i}, y_i, g^{u_j}\right)$, where $i \in [1, N]$ and $j \in [1, U]$.

(2) Key Generation

1) *IdKeyGen*: DU is assigned a unique identifier *uid* and a set of attributes $S_{uid}$ when it joins the BEM-ABSE, and then DU randomly selects $z \in Z_p^*$ and calculates $g^{a/z}$ and $g^{1/z}$. After that, DU sends its public identity key $upk_{uid} = \left(g^{a/z}, g^{1/z}\right)$ to the BC registration and keeps the private identity key $usk_{uid} = z$.

2) *SKGen*: When the DU is successfully registered, each $AA_i$ associated with the attribute in $S_{uid}$ generates the decryption key $SK_{i,uid}$ for the DU *uid*. Attribute authority input the $S_{uid} = \left\{a_j\right\}$, $n_{uid} = |S_{uid}|$, $u = H\left(uid\right)$ and selects $t_i \in Z_p^*$. After that, $AA_i$ perform the following calculation $SK_{i,uid}$ with Eq. (5). Finally, $AA_i$ sends the $SK_{i,uid}$ to DU for constructing the user transform key $TK_{uid}$.

$$SK_{i,uid} = \left\{K_i = g^{au/z} g^{at_i}, L_i = g^{u/z} g^{t_i/z}, L_{i,j} = g^{u\left(a_j - u_j\right)/z} g^{\left(a_j - u_j\right)t_i/z}\right\}_{i \in u_A, j \in n_{uid}} \tag{5}$$

(3) Encryption

1) *Offline.Enc*: DO perform offline encryption on computationally capable devices before determining the access structure and extracting keywords. First, DO selects $\lambda_j^{'}, u_j^{'}, r_j \in Z_p^*$, where $j \in [1, U]$. Then DO calculates $C_{1,j} = g^{a\lambda_j^{'}} g^{u'(-r_j)} g^{u_j r_j}$ and $C_{2,j} = g^{r_j}$. Finally, DO outputs an intermediate ciphertext $IC = \left\{C_{1,j}, C_{2,j}, \lambda_j^{'}, u_j^{'}\right\}_{j \in [1,U]}$.

Leveraging *IC* on end devices with limited resources, such as sensors and wearables, can help decrease the processing overhead of the encryption process. In addition, the *IC* can be used multiple times when the attributes owned by the user remain unchanged.

2) *Online.Enc*: After obtaining the intermediate ciphertext *IC*. Firstly, DO chooses a random number $m \in G_T$ and calculates $K = H_2\left(m\right)$ as the symmetric key, and then DO generates the ciphertext $C_F = Enc_{sym}\left(K, F\right)$ and verification value $VK_F = H_1\left(H_2\left(m\right) || C_F\right)$ of the data file $F$.

Then, the DO protects $m$ with the specified access policy $(M_{l \times n}, \rho)$, where $M_{l \times n}$ is a matrix with $l$ rows and $n$ columns, the function $\rho$ maps each row of $M_{l \times n}$ to an attribute. The DO chooses a vector $v = (s, y_2, y_3 \cdots, y_n)$ and calculates the $\lambda_j = M_j v$, where $y_2, y_3 \cdots, y_n \in Z_p^*$ is used to share the encryption element $s$ and $M_j$ refers to the $j$-th row of the matrix $M_{l \times n}$. After that, DO calculates $C = m \prod_{i \in u_A} e\left(g,g\right)^{\alpha_i s}$, $C' = g^s$, $C_{3,j} = \lambda_j - \lambda_j^{'}$ and $C_{4,j} = u_j^{'} - \rho\left(j\right)$, where $j \in [1, l]$ and $u_A$ is the set of associated attribute authorities, then DO outputs $CT = \left(C, C', C_{1,j}, C_{2,j}, C_{3,j}, C_{4,j}, C_F, (M_{l \times n}, \rho), VK_F\right)$.

Next, the DO extracts keywords set $\mathcal{W} = (w_1, w_2, \cdots, w_n)$ from $F$ and randomly selects elements $\xi_i \in Z_p^*$ for each keyword $w_i \in \mathcal{W}$. After that, it calculates the keywords index set $I_{\mathcal{W}} = \left\{I_{w_i}\right\}_{w_i \in \mathcal{W}} = \left\{\left[I_{1,w_i}, I_{2,w_i}\right]\right\}$, where $I_{1,w_i} = g^{\xi_i}$ and $I_{2,w_i} = H_2\left(e\left(\left(g^\gamma\right)^{\xi_i}, H_1\left(w_i\right)\right)\right)$. Finally, DO sends the $(CT, I_{\mathcal{W}})$ to

the nearest EN, and then DO stores the $CT$ on CSP with the address *address* and submits the index storage transaction with embedded $I_{\mathcal{W}}$ and *address* to the blockchain.

(4) Trapdoor Generation

*TrapGen*: When a DU searches for data files according to his keyword $w'$ of interest, DU first selects a random element $\delta \in Z_p^*$ and inputs $GP$, then DU calculates $T_{1,w'} = H_2\left(e\left(g^{\gamma}, (g^c)^{\delta}\right)\right)$, $T_{2,w'} = g^{\delta}$ ,$T_{3,w'} = H_1(w')$. Finally, DU embeds $T_{w'} = \left(T_{1,w'}, T_{2,w'}, T_{3,w'}\right)$ into the generated search transaction and submits it to the BC.

(5) Search

*Search*: After receiving the search transaction from DU, the SSC checks whether $T_{w'}$ matches index $I_w$ with Eq. (6), where $\theta = \left(T_{3,w'}\right)^{\gamma} \oplus T_{1,w'}$ and $\varphi = H_1\left(e\left(\left(T_{2,w'}\right)^c, g^{\gamma}\right)\right)$ are generated by blockchain nodes executing the Pedersen secret sharing protocol. If the above condition both holds, SSC returns the search result relevant ciphertext address *address* to DU. Otherwise, it returns $\perp$.

$$H_2(e(I_1, \theta \oplus \varphi)) = I_2 \tag{6}$$

(6) Decryption

1) *EN.Dec*: When EN receives the $TK_{uid}$ and the ciphertext $CT$ obtained from CSP using the ciphertext address from DU, it performs ciphertext transform for $CT$. If the DU's attributes set $S_{uid}$ satisfies the access policy $(M_{l\times n}, \rho)$ embedded in ciphertext $CT$, and let's define the mapping of user attributes as $I = \{j : \rho(j) \in S_{uid}\}$ where $I \subseteq \{1, 2, \cdots, l\}$, there must exist a collection of constants $\left\{o_j \in Z_p^*\right\}$ such that $\sum_{j\in S_{uid}} o_j\lambda_j = s$, where $\lambda_j = M_j v$. Then, DU further obtains $CT'$ with Eq. (7). Finally, the EN returns $(C, CT', C_F)$ back to DU.

$$CT' = \prod_{i\in u_{\mathrm{A}}} \frac{e(C', K_i)}{\prod_{j\in I, \rho(j)\in S_i} e\left(C_j, L_i\right) e\left(C_{2,j}, L_{i,j}\right)} = \prod_{i\in u_{\mathrm{A}}} e(g, g)^{\alpha_i s/z} \tag{7}$$

2) *User.*Dec: After receiving the transform ciphertext from EN, the DU utilizes its private key $usk_{uid}$ to decrypt and retrieve the random number $m = C/(CT')^z$. Then, DU generates a validation transaction and embeds $m$ and *address* in it before submitting it to the BC to verify the equality relationship between $H_1(H_2(m)||C_F)$ and $VK_F$ through the VSC. If yes, the DU obtains the complete outsourced decrypted data and decrypts the data file $F = Dec_{sym}(K, C_F)$ with the symmetric key $K = H_2(m)$. Otherwise, decryption fails and outputs $\perp$. It is worth noting that data validation is not mandatory during the decryption process.

## 6 Security Analysis

***Theorem 1*:** If the decisional $q$-BDHE assumption holds, the BEM-ABSE scheme achieves IND-CPA security.

***Proof*:** Assume there is a game that can be won in PPT by the adversary $\mathcal{A}$ with a non-negligible advantage $\varepsilon$. Then, we construct a simulator $\mathcal{B}$ with a non-negligible advantage $\varepsilon/2$ to solve the decisional $q$-BDHE problem. The simulation is carried out as follows.

***Init*:** $\mathcal{B}$ receives a $q$-BDHE challenge instance $(y, T)$. $\mathcal{A}$ chooses the access structure $\left(M_{l\times n}^*, \rho^*\right)$ and $S_{\mathrm{A}}' \in S_{\mathrm{A}}$ is a set of corrupted attribute authority, where $M_{l\times n}^*$ has $l < q$ columns.

***Setup*:** $\mathcal{B}$ chooses a security parameter $\lambda$, then uses the Pedersen algorithm to perform the *GlobalSetup* and produce the global parameter $GP$. Each uncorrupted $\mathrm{AA}_i$ executes the

*AuthoritySetup* algorithm where $i \in \left(S_{\mathrm{A}} - S_{\mathrm{A}}^{'}\right)$. Then, $\mathcal{B}$ picks a random element $\alpha_i^{'} \in Z_p$ and implicitly lets $\alpha = \alpha_i^{'} + a^{q+1}$ by setting $e(g,g)^{\alpha_i} = e(g,g)^{\alpha_i^{'}} e\left(g^a, g^{a^q}\right), y_i = g^{\alpha_i} = g^{\alpha_i^{'}+a^{q+1}}$. For each attribute $a_j$, choose an element $z_j \in Z_p$ at random, then calculate $g^{u_j} = g^{z_j} \prod_{i \in X} g^{aM_{x,1}^*/b_x} g^{a^2 M_{x,2}^*/b_x} \cdots g^{nM_{x,n}^*/b_x}$, where implicitly defines $u_j$ as $z_j + \sum_{x \in X} \dfrac{aM_{x,1}^*}{b_x} + \dfrac{a^2 M_{x,2}^*}{b_x} + \cdots + \dfrac{a^n M_{x,n}^*}{b_x}$, note that if $X = \varnothing$ then $g^{u_j} = g^{z_j}$. Finally, $\mathcal{B}$ returns $PK_i$ back to $\mathcal{A}$.

***Phase* 1:** $\mathcal{A}$ sends $(S, S_{uid}, uid)$ to $\mathcal{B}$ for the decryption key $SK_{i,uid}$ query with the user attribute set $S_{uid}$, global set of attributes $S = \{S_i\}_{i \in S_{\mathrm{A}} - S_{\mathrm{A}}^{'}}$ and user identification $uid$. $\mathcal{B}$ random chooses element $d_i \in Z_p$ and column vector $\omega = (\omega_1, \omega_2, \cdots, \omega_n)^{\mathrm{T}}$, such that $\omega_1 = -1$ and $M_l^* \omega = 0$ for all $i$ where $\rho^*(i) \in S_{uid}$. $\mathcal{B}$ defines $t_i = d_i + \omega_1 a^q + \omega_2 a^{q-1} + \cdots + \omega_n a^{q-n+1}$ and computes $L_i = g^{u+t_i/z} = g^{u+d_i/z} \prod_{k=0}^{n} \left(g^{a^{q+1-k}}\right)^{\omega_k/z}$. Due to $\omega_1 = -1$, $g^{at_i}$ contains the factor $g^{-a^{q+1}}$, but this portion can be cancelled out by a factor in $g^{\alpha_i}$, allowing $\mathcal{B}$ to calculate $K_i = g^{au/z} g^{\alpha_i^{'}/z} g^{ad_i/z} \prod_{k=2}^{n} \left(g^{a^{q+1-k}}\right)^{\omega_k/z}$. For each attribute $a_j \in S_{uid}$, if it exists $\rho^*(x) = a_j$, let $L_{i,j} = L_i^{a_j - z_j}$, otherwise, $L_{i,j} = L_i^{a_j - z_j} \prod_{x \in X} \prod_{k=1}^{n} \left(\left(g^{a^k}\right)^{(u+d_i)/z} \prod_{f=1, f \neq k}^{n} \left(g^{a^{q+1+f-k}/b_x}\right)^{\omega_k/z}\right)^{-M_{x,k}^*}$. Finally, $\mathcal{B}$ returns $SK_{S,uid} = \{K, L, L\}_{i \in S_{\mathrm{A}} - S_{\mathrm{A}}^{'}, j \in S}$ back to $\mathcal{A}$.

***Challenge*:** $\mathcal{A}$ submits two challenge messages $m_0^*, m_1^*$ with equal length. Then, $\mathcal{B}$ randomly selects bit $b$ and recovers $m_b^*$ under $(M^*, \rho^*)$. After that, $\mathcal{B}$ computes $C = m_b^* Te\left(g^s, g^{\alpha_i^{'}}\right)$. Then, $\mathcal{B}$ constructs a vector $v = \left(s, sa + y_2, sa^2 + y_3, \cdots, sa^{n-1} y_n\right)$ for achieving secret sharing of the $s$, where each element $y_2, \cdots, y_n \in Z_p$ in $v$ is randomly chosen. Since there exists $\lambda' = M_{l \times n}^* v$, it is possible to construct $\lambda_j^{'} = \sum_{k=1}^{n} M_{j,k}^* sa^{k-1} + \widetilde{\lambda}_j$ from the vector $v$ with $\widetilde{\lambda}_j = \sum_{k=2}^{n} M_{j,k}^* sy_k$. $\mathcal{B}$ randomly selects an element $z_j^{'} \in Z_p$ and calculates $u_j^{'} = z_j^{'} + \sum_{x \in X} \dfrac{aM_{x,1}^*}{b_x} + \dfrac{a^2 M_{x,2}^*}{b_x} + \cdots + \dfrac{a^n M_{x,n}^*}{b_x}$. For $i = 1, 2, \cdots, l$, $\mathcal{B}$ first selects random elements $r_j^{'}, \beta_j, \gamma_j \in Z_p$ and defines $\gamma_j = -\left(r_j^{'} + sb_j\right)$, then calculates $C_{2,j} = g^{r_j} = g^{-r_j^{'} - sb_j}, C_{3,j} = \beta_j$, $C_{4,j} = \gamma_j$ and $C_{1,j} = g^{a\lambda_j^{'}} g^{-u_j^{'} r_j} g^{u_j r_j} = g^{a\widetilde{\lambda}_j} g^{-\left(r_j^{'} + sb_j\right)\left(z_j - z_j^{'}\right)} \prod_{k=1}^{n} \left(g^{sa^k}\right)^{M_{j,k}^*}$. Finally, $\mathcal{B}$ returns $CT^*$ back to $\mathcal{A}$.

***Phase* 2:** $\mathcal{A}$ adaptively repeats the execution of query *Phase* 1, while it should follow the constraints of the query phase.

***Guess*:** $\mathcal{A}$ outputs its guessed bit $b'$, and if $b = b'$, $\mathcal{A}$ wins the attack game; Otherwise, it fails. If $\eta = 0$, $\mathcal{B}$ guess $T = e(g,g)^{a^{q+1} s}$, $\mathcal{A}$ obtains the legitimate ciphertext of $m_b^*$ and gains the game with the probability $\varepsilon = \Pr[b = b'] - \dfrac{1}{2}$, the probability that $\mathcal{B}$ wins is $\varepsilon = \Pr[b = b' | \eta = 0] = \Pr[b = b'] = \varepsilon + \dfrac{1}{2}$. If $\eta = 1$, there is a random element in the ciphertext, the $\mathcal{B}$ wins the game with probability is $\varepsilon = \Pr[b = b' | \eta = 1] = \Pr[b \neq b'] = \dfrac{1}{2}$. Thus, the probability of $\mathcal{B}$ solving the q-BDHE problem is $Adv_{\mathrm{A}}^{IND-CPA} = \left|\Pr[b = b'] - \dfrac{1}{2}\right| = \dfrac{\varepsilon}{2}$.

Because of the hardness of the $q$-BDHE problem, the advantage $Adv_{\mathrm{A}}^{IND-CPA} = \dfrac{\varepsilon}{2}$ of the adversary in breaking the BEM-ABSE scheme is negligible.

***Theorem 2*:** If the bilinear Diffie-Hellman assumption holds, the BEM-ABSE scheme achieves IND-CKA security.

***Proof*:** Assume there is a game that can be won in PPT by the adversary $\mathcal{A}$ with a non-negligible advantage $\varepsilon$. Then, we construct a simulator $\mathcal{B}$ with a non-negligible advantage $\varepsilon/eq_{H_2}q_T$ to solve the bilinear Diffie-Hellman problem, let $e$ be the base of the natural logarithm, $q_{H_2}$ and $q_T$ denote the maximum query limits for the hash function $H_2$ and trapdoor, respectively.

***Init*:** Assume that given a BDH tuple $(g, u_1 = g^{\delta_1}, u_2 = g^{\delta_2}, u_3 = g^{\delta_3})$, where $\delta_1, \delta_2, \delta_3 \in Z_p$ are random elements, the objective of $\mathcal{B}$ is to calculate $e(g,g)^{\delta_1\delta_2\delta_3} \in G_T$.

***Setup*:** $\mathcal{B}$ chooses a security parameter $\lambda$ and invokes the trusted node to execute the Pedersen algorithm to obtain the $g^c$ and $g^\gamma$ with the input $g^{\delta_1}$, where $g^\gamma = g^{\delta_1 t_1} = u_1^{t_1}$. Then, $\mathcal{B}$ returns public parameters $(H_1, H_2, g^c, g^\gamma)$ to the adversary $\mathcal{A}$ that are only relevant to conducting a keyword search.

***Phase* 1:** $\mathcal{A}$ can adaptively issue the subsequent oracles in PPT.

$\mathcal{O}_{H_1}(w_i)$: The $\mathcal{B}$ first initializes a hash list $L_{H_1}$ of tuples $(w_i, h_i, a_i, c_i, )$. While adversary queries $H_1$ with a specific keyword $w_i$, $\mathcal{B}$ first search the $L_{H_1}$ list. If $w_i$ already exists in $L_{H_1}$, it returns $h_i$ back to $\mathcal{A}$. Alternatively, $\mathcal{B}$ randomly selects a bit $c_i \in \{0,1\}$ where $\Pr[c_i = 0] = 1/(q_T + 1)$. After that, $\mathcal{B}$ selects a random element $a_i \in Z_p$ and calculates $h_i$. Notice that when $c_i = 0$, $h_i = u_2^{a_i} \in G$; Otherwise, $h_i = g^{a_i} \in G$. Finally, $\mathcal{B}$ returns $h_i$ back to $\mathcal{A}$ while storing $(w_i, h_i, a_i, c_i, )$ in the list $L_{H_1}$.

$\mathcal{O}_{H_2}(t_i)$: The $\mathcal{B}$ initializes list $L_{H_2}$ for the tuple $(t_i, v_i)$. $\mathcal{A}$ queries $H_2$ with an arbitrary element $t_i \in G_T$, and if $t_i$ has been previously queried, $\mathcal{B}$ searches $L_{H_2}$ for the associated query result and returns it to $\mathcal{A}$. Otherwise, $\mathcal{B}$ randomly select $v_i \in \{0,1\}^{\log p}$ and sends $H_2(t_i) = v_i$ to $\mathcal{A}$ while storing $(t_i, v_i)$ in the list $L_{H_2}$.

$\mathcal{O}_{T_w}(w_i)$: When $\mathcal{A}$ issues a trapdoor query with the keyword $w_i$, $\mathcal{B}$ first queries list $L_{H_1}$ to obtain $H_1(w_i) = h_i$ and the corresponding tuple $(w_i, h_i, a_i, c_i, )$. Notice that when $c_i = 0$, $\mathcal{B}$ terminates the query. Otherwise, $h_i = g^{a_i} \in G$. Then, $\mathcal{B}$ randomly selects an element $\xi \in Z_p$ to query $\mathcal{O}_{H_2}(t_i)$ to get $H_2\left(e\left(g^c, (g^\gamma)^\xi\right)\right)$ and calculates $T^*_{1,w_i} = H_2\left(e\left(g^c, (g^\gamma)^\xi\right)\right), T^*_{2,w_i} = g^\xi$ and $T^*_{3,w_i} == H_1(w_i) = g^{a_i}$. Finally, $\mathcal{B}$ sends $T^*_{w_i}$ to $\mathcal{A}$.

***Challenge*:** $\mathcal{A}$ provides challenge query keywords $w_0^*$ and $w_1^*$, where the lengths of keywords are the same. After receiving the challenge keywords, $\mathcal{B}$ generates the index by performing the following steps: $\mathcal{B}$ first obtains $H_1(w_1) = h_1$ and $H_2(w_2) = h_2$ form $\mathcal{O}_{H_1}(w_i)$ and the associated $(w_i, h_i, a_i, c_i, )_{i\in\{0,1\}}$ by retrieving $L_{H_1}$. Notice that $\mathcal{B}$ will terminate the current challenge if both $c_0$ and $c_1$ are either 0 or 1. Otherwise, $\mathcal{B}$ randomly selects a bit $b$ such that $c_b = 0$. $\mathcal{B}$ chooses a random number $t_2 \in Z_p$ and $J \in \{0,1\}^{\log p}$, and then calculates the challenge keyword index $I^*_{w_b} = \left(I^*_{1,w_b}, I^*_{2,w_b}\right) = \left((u_3)^{1/t_2}, J\right)$ with the implicit defines $\xi = \delta_3/t_2$ and $J = H_2\left(e\left((g^\gamma)^\xi, H_1(w_b)\right)\right) = H_2\left(e(g,g)^{\delta_1\delta_2\delta_3(a_i t_1/t_2)}\right)$. We can know that $I^*_{w_b}$ is a valid index of $w_b$ as required. Finally, $\mathcal{B}$ returns $I^*_{w_b}$ back to the adversary.

***Phase* 2:** $\mathcal{A}$ adaptively repeats the execution of query *Phase* 1, while it should follow the constraints of the query phase.

***Guess*:** $\mathcal{B}$ takes any tuple $(t_i, v_i)$ from $L_{H_2}$ and outputs $t^{a_b t_1/t_2}$ as guess form $e(g,g)^{\delta_1\delta_2\delta_3}$. In the following, we analyze the probability of $\mathcal{B}$ correctly outputting $e(g,g)^{\delta_1\delta_2\delta_3}$. It is known that the probability of $\mathcal{B}$ terminating execution is at most $1/ep_T$ in the simulation phase, and the probability of $\mathcal{A}$ performing a query $H_2\left(e\left(u_3^\xi, H_1(w_0)\right)\right)$ or $H_2\left(e\left(u_3^\xi, H_1(w_1)\right)\right)$ is at least $2\varepsilon$ in the attack phase [9]. In other words, $\mathcal{A}$ submits $e\left(u_3^\xi, H_1(w_b)\right) = e(g,g)^{\delta_1\delta_2\delta_3(a_i t_1/t_2)}$ to execute $\mathcal{O}_{H_2}$ queries with probability at least $\varepsilon$ and $\mathcal{B}$ correctly selects the associated tuple with probability at least $1/q_{H_2}$. Therefore, $\mathcal{B}$ has a

success probability of at least $\varepsilon/q_{H_2}$ in outputting the correct result. In fact, $\mathcal{B}$ would be able to succeed with a probability of at least $\varepsilon/eq_{H_2}q_T$.

## 7 Performance Analysis

### *7.1 Functional Analysis*

BEM-ABSE supports multi-authority, LSSS, on/offline encryption, assisted decryption, results verification and blockchain. The functional features are compared in Table 1. Schemes [29,31] and BEM-ABSE are all SE schemes based on multiple authorization centers. The access policy is based on LSSS, which can effectively avoid single-point failures and improve the system's security. However, A significant computational cost is placed on the client by other systems, with the exception of the BEM-ABSE scheme, which does not have the design of online/offline procedures or edge-assisted decryption throughout the encryption and decryption stages. Despite the fact that the scheme [29] outsources encryption and decryption to save costs for the client, its overall computing cost is significant, and its impact is poor. Schemes [20] and BEM-ABSE support data integrity verification. Furthermore, BEM-ABSE runs ciphertext search via a smart contract and uploads the ciphertext index to the blockchain, which can better safeguard user privacy and data security.

**Table 1:** Functional comparison

| Scheme | Multi-authority | Access structure | On/offline encryption | Assisted decryption | Results verification | Blockchain |
|---|---|---|---|---|---|---|
| [10] | × | Tree | × | × | × | × |
| [20] | × | Tree | × | × | √ | √ |
| [29] | √ | LSSS | √ | √ | × | × |
| [31] | √ | LSSS | × | √ | × | × |
| BEM-ABSE | √ | LSSS | √ | √ | √ | √ |

### *7.2 Theoretical Analysis*

In theoretical computations, the computational complexity is primarily evaluated by considering the pairing $P$ and the exponentiation $E\,(E_T)$ operations on the group $G\,(G_T)$. Multiplication and hash operations are relatively lighter in comparison and are not given as much emphasis in terms of computational analysis.

The computational complexity of the selected method was analyzed, and a detailed study was conducted on the differences in computational costs. The results were compared in Table 2, where $|S|$ is the number of attributes of the user and $l$ is the number of attributes in the access policy. As the number of attributes increases linearly, in comparison to the other two, BEM-ABSE generates keys at a lower computational cost. Due to the online/offline strategy used in encryption, the computational overhead of DO online encryption is $3E+E_T+P$, while the computational costs for trapdoor creation and search are unaffected. In the decryption phase, due to the use of EN assistance for decryption, DU's computational cost is $E_T$. Note that in the table, $\Delta$ represents $2l+1$ and $\Theta$ represents $2\,|S|+1$ and "—" represents without consideration.

The storage cost comparison results of these schemes are shown in Table 3. Where $|G|$, $|G_T|$ and $|Z_p|$ are used to specify the lengths of elements $G$, $G_T$, and $Z_p$, respectively. The quantity of attributes

influences the size of the user's key and ciphertext. In comparison to methods [20] and [29], BEM-ABSE has less storage overhead during the key generation and ciphertext generation phases. It is worth noting that the storage cost in the trapdoor generation and search phase is constant, which has a significant advantage over the other two schemes.

**Table 2:** Comparison of computational cost

| Scheme | Key generation | Encryption | Trapdoor generation | Search | Decryption |
|---|---|---|---|---|---|
| [29] | $(5\|S\|+10)E$ | $(3l+3)E_T+7lE+2P$ | $(\|S\|+3)E$ | $2lP+\Delta E+lE_T$ | $\Delta P+(l+2)E+3lE_T$ |
| [20] | $(3\|S\|+1)E$ | $(\Delta+5)E+P$ | $(2\|S\|+1)E+P$ | $(\Delta+2)P+E$ | — |
| BEM-ABSE | $(2\|S\|+5)E$ | $(4l+3)E+E_T+P$ | $2E+P$ | $2E+2P$ | $\Delta P+E_T$ |

**Table 3:** Comparison of storage cost

| Scheme | Key generation | Encryption | Trapdoor generation | Search | Decryption |
|---|---|---|---|---|---|
| [29] | $\|S\|\|G\|$ | $5l\|G\|+4l\|Z_p\|+\Delta\|G_T\|$ | $(\|S\|+3)\|G\|+\|Z_p\|$ | $\Delta\|G_T\|+3\|G\|$ | $3(l+1)\|G_T\|+2l\|G\|$ |
| [20] | $\Theta\|G\|+(\|S\|+1)\|Z_p\|$ | $(2l+4)\|G\|+2\|Z_p\|$ | $(\Theta+2)\|G\|+\|Z_p\|$ | $5\|G_T\|$ | — |
| BEM-ABSE | $(\|S\|+2)\|G\|$ | $(\Delta+1)\|G\|+2l\|Z_p\|+\|G_T\|$ | $2\|G\|+\|G_T\|$ | $2\|G\|+2\|G_T\|$ | $\Delta\|G_T\|$ |

### *7.3 Experimental Analysis*

The experiment simulated the deployment of a Hyperledger Fabric on a server with an Inter® Xeon® E5-2630 CPU @2.3 GHz 16-core and 64 GB RAM. We instantiated an edge node on a laptop with a 2.8 GHz Intel® Core™ i7-1165 and 16 GB of RAM and instantiated a resource-constrained device on a Raspberry Pi 3B with a Quad-Core ARMv8 CPU @1.2 GHz 4-core processor and 1 GB of RAM. The Fabric network is made up of three order nodes and four peer nodes that use the Raft consensus mechanism. Note that the experiment used the Pairing-Based Cryptography Library (PBC) to implement cryptographic operations and chose an elliptic curve group with type A: $y^2=x^3+x$ and the order of the group is 160 bits. When the $G$ ($G_T$) group order is set as 512 bits, we can obtain $|Z_p|$ with a length of 160 bits and $|G|$ and $|G_T|$ with a length of 1024 bits. Moreover, we also set $l=|S|\in[0,50]$.

Fig. 2 describes the computation and storage cost of BEM-ABSE. The comparison of computation costs is given in Figs. 2a–2d. In Fig. 2a, we noticed that the time cost for all three methods has a direct correlation with the number of attributes during the key generation procedure. Notably, when compared to the other two systems, the BEM-ABSE method has reduced computing costs. In Fig. 2b, scheme BEM-ABSE adopts an online/offline encryption mechanism. Note that although the BEM-ABSE scheme's computational cost during the encryption phase is larger than that of the scheme [20], the BEM-ABSE scheme uses intermediate ciphertexts for online encryption during DO usage in the encryption process rather than performing offline encryption every time during encryption.

When we set $l$=50, the time cost of DO online encryption is 25.52 ms. The computational costs for the trapdoor generation and search are shown in Figs. 2c and 2d, respectively. In the BEM-ABSE scheme, the computational costs of trapdoor generation and ciphertext search remain constant.

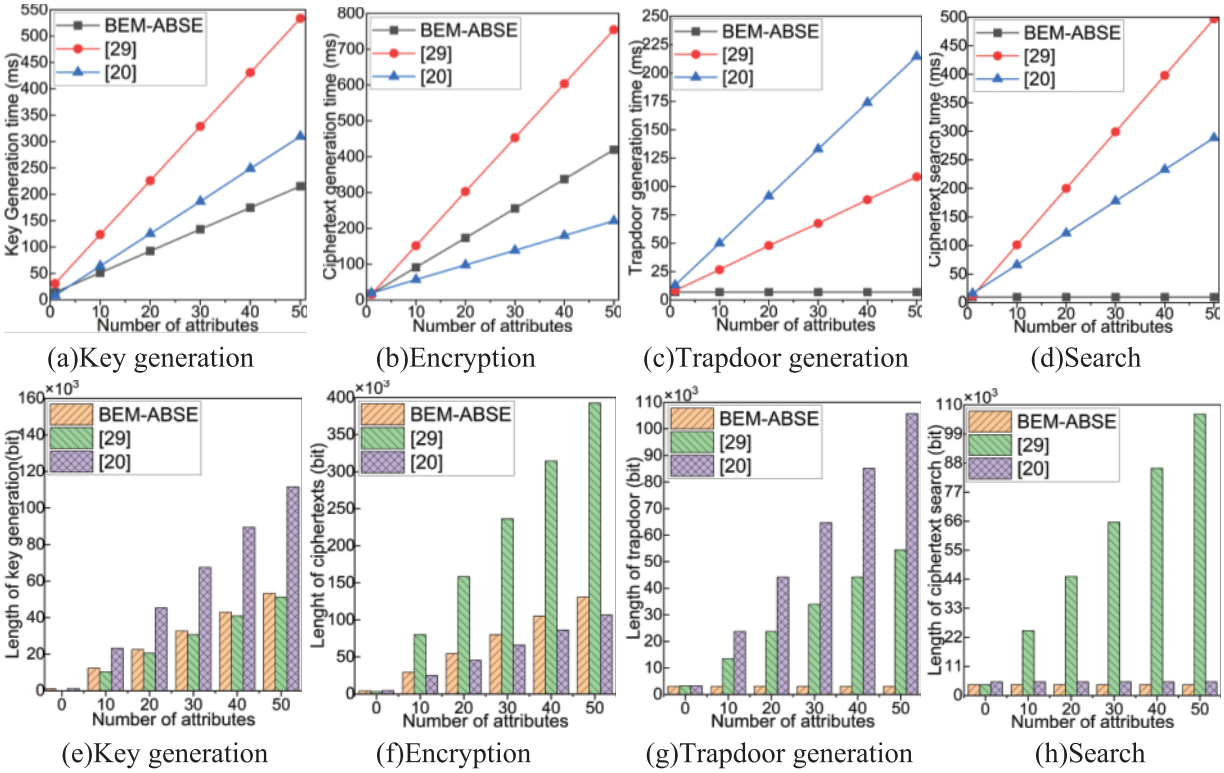


**Figure 2:** Algorithm time and storage cost

Next, the comparison of storage costs is given in Figs. 2e and 2h. As illustrated in Fig. 2e, the key generation stage storage costs in schemes [29] and BEM-ABSE are similar. When the number of attributes reaches 50, the scheme [20] has a storage cost that is almost double that of the BEM-ABSE. In Fig. 2f, it can be observed that in the encryption stage, the storage cost of the scheme [29] escalates significantly as the number of attributes increases, surpassing the storage cost of the schemes [20] and BEM-ABSE by a significant margin. Figs. 2g and 2h demonstrate that the storage costs associated with trapdoor generation and search stages in BA-ABSE are denoted as $2|G| + |G_T|$ (0.38KB) and $2|G| + 2|G_T|$ (0.5KB), respectively, and remain unaffected by the number of attributes. However, the storage costs in the same stage for schemes [20] and [29] increase linearly with the attribute.

In Fig. 3, the decryption time overhead is depicted. It can be observed that both the BME-ABSE scheme and [29] exhibit a linear increase in decryption time overhead as the number of attributes in the ciphertext policy grows. The BME-ABSE scheme has a total decryption time of 282 ms when there are 50 attributes which is much less than the 423 ms of the scheme [29]. In order to further lower the DU's computing expense during the decryption stage, the BME-ABSE scheme delegates the task of converting ciphertext with higher computational cost to ENs. At the same time, DU only needs to perform consistent operations regardless of the access policy. The utilization of computational resources on ENs simplifies the decryption process, reduces complexity, and shortens the time cost

of decryption. In order to increase the decryption efficiency of IoT devices with limited resources, a lightweight decryption procedure is advantageous.

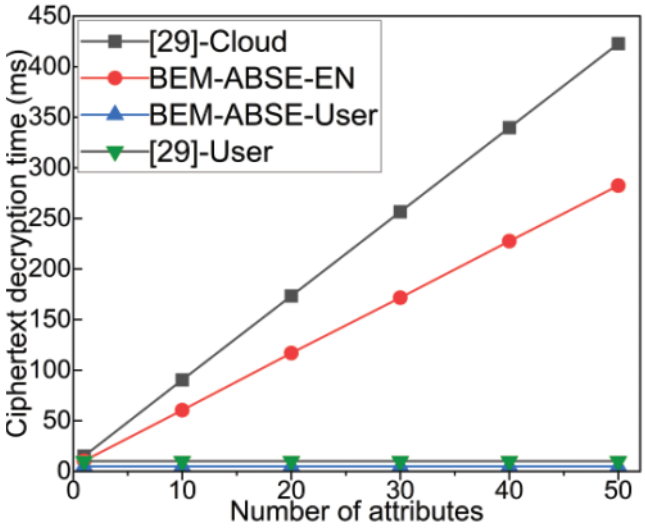


**Figure 3:** Decryption time cost

## 8 Conclusion

This paper presents an efficient multi-authority attribute-based searchable encryption scheme with blockchain assistance (BEM-ABSE) for cloud-edge collaborative scenarios. The BEM-ABSE scheme introduces an online/offline hybrid encryption mechanism. It adopts an edge-assisted outsourcing decryption mechanism, significantly improving the efficiency of encryption and decryption and effectively reducing the computation overhead of resource-limited IoT devices. The consortium blockchain serves as a trusted authentication center for global parameter generation and management, and the introduction of smart contracts realizes trusted and fair ciphertext keyword search and decryption result verification. BEM-ABSE has been rigorously analyzed for security and shown to be secure against IND-CPA and IND-CKA attacks. Performance analysis confirms its efficiency and practicality. However, a major limitation of the BEM-ABSE is its lack of support for expressive search queries such as fuzzy search and multi-keyword search and its inability to revoke permissions for malicious users. Future work will focus on designing a flexible indexing and efficient permission revocation scheme, enabling the BEM-ABSE to support various controllable search requests.

**Acknowledgement:** We thank the anonymous reviewers and editors for their very constructive comments.

**Funding Statement:** This work is supported by the National Natural Science Foundation of China (Nos. 62162018, 61972412), the Natural Science Foundation of Guangxi (No. 2019GXNSFGA245004), the Guilin Science and Technology Project (20210226-1) and the Innovation Project of Guangxi Graduate Education (No. YCSW2022296).

**Author Contributions:** The authors confirm contribution to the paper as follows: study conception and design: Peng Liu, Qian He; data collection: Peng Liu; analysis and interpretation of results: Peng Liu,

Biao Guo; draft manuscript preparation: Peng Liu. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** The data used to support the findings of this study are included within the article.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References


[1] K. Yu, J. Yu and C. Luo, "The impact of mobility on physical layer security of 5G IoT networks," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 1–14, 2022.

[2] S. Jiang, J. Cao, H. Wu, K. Chen and X. Liu, "Privacy-preserving and efficient data sharing for blockchain-based intelligent transportation systems," *Information Sciences*, vol. 635, pp. 72–85, 2023.

[3] Y. Chiang, Y. Zhang, H. Luo, T. Y. Chen, G. H. Chen *et al.,* "Management and orchestration of edge computing for IoT: A comprehensive survey," *IEEE Internet of Things Journal*, vol. 1, pp. 1–30, 2023.

[4] L. Zhang, H. Xiong, Q. Huang, J. G. Li, K. K. R. Choo *et al.,* "Cryptographic solutions for cloud storage: Challenges and research opportunities," *IEEE Transactions on Services Computing*, vol. 15, no. 1, pp. 567–587, 2022.

[5] J. Liu, Y. Li, R. Sun, Q. Pei, N. Zhang *et al.,* "EMK-ABSE: Efficient multikeyword attribute-based searchable encryption scheme through cloud-edge coordination," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 18650–18662, 2022.

[6] F. Guo, Y. Mu, W. Susilo, W. Hsing, D. S. Wong *et al.,* "Optimized identity-based encryption from bilinear pairing for lightweight devices," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 6, pp. 211–220, 2015.

[7] V. Goyal, O. Pandey, A. Sahai and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. of the 13th ACM Conf. on Computer and Communications Security*, New York, NY, USA, pp. 89–98, 2006.

[8] D. X. Song, D. Wagner and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. of the 2000 IEEE Symp. on Security and Privacy*, Washington DC, USA, pp. 44–55, 2000.

[9] Q. Zheng, S. Xu and G. Ateniese, "VABKS: Verifiable attribute-based keyword search over outsourced encrypted data," in *Proc. of the IEEE INFOCOM 2014*, Toronto, ON, Canada, pp. 522–530, 2014.

[10] Y. Miao, X. Liu, K. K. R. Choo, R. H. Deng, J. Li *et al.,* "Privacy-preserving attribute-based keyword search in shared multi-owner setting," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1080–1094, 2021.

[11] M. Ali, M. R. Sadeghi, X. Liu, Y. Miao and A. V. Vasilakos, "Verifiable online/offline multi-keyword search for cloud-assisted industrial internet of things," *Journal of Information Security and Applications*, vol. 65, pp. 103101, 2022.

[12] Y. Miao, Q. Tong, R. H. Deng, K. K. R. Choo, X. Liu *et al.,* "Verifiable searchable encryption framework against insider keyword-guessing attack in cloud storage," *IEEE Transactions on Cloud Computing*, vol. 10, no. 2, pp. 835–848, 2020.

[13] X. Xiang and X. Zhao, "Blockchain-assisted searchable attribute-based encryption for e-health systems," *Journal of Systems Architecture*, vol. 124, pp. 102417, 2022.

[14] M. Zhang, J. Cao, Y. Sahni, Q. Chen, S. Jiang *et al.,* "Blockchain-based collaborative edge intelligence for trustworthy and real-time video surveillance," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1623–1633, 2022.

[15] Q. Huang, Q. Wei, G. Yan, L. Zou and Y. Yang, "Fast and privacy-preserving attribute-based keyword search in cloud document services," *IEEE Transactions on Services Computing*, vol. 1, pp. 1–13, 2023.

[16] K. Zhang, J. Long, X. Wang, H. N. Dai, K. Liang *et al.,* "Lightweight searchable encryption protocol for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4248–4259, 2020.
[17] Y. Miao, Q. Tong, K. K. R. Choo, X. Liu, R. H. Deng *et al.,* "Secure online/offline data sharing framework for cloud-assisted industrial internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8681–8691, 2019.
[18] R. Zhou, X. Zhang, X. Wang, G. Yang, H. N. Dai *et al.,* "Device-oriented keyword-searchable encryption scheme for cloud-assisted industrial IoT," *IEEE Internet of Things Journal*, vol. 9, no. 18, pp. 17098–17109, 2021.
[19] X. Yang, G. Chen, M. Wang, T. Li and C. Wang, "Multi-keyword certificateless searchable public key authenticated encryption scheme based on blockchain," *IEEE Access*, vol. 8, pp. 158765–158777, 2020.
[20] S. Niu, M. Song, L. Fang, S. Han and C. Wang, "Keyword search over encrypted cloud data based on blockchain in smart medical applications," *Computer Communications*, vol. 192, pp. 33–47, 2022.
[21] M. Chase, "Multi-authority attribute-based encryption," in *Proc. of the TCC 2007*, Amsterdam, Netherlands, pp. 515–534, 2007.
[22] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proc. of the EUROCRYPT 2011*, Tallinn, Estonia, pp. 568–588, 2011.
[23] S. Tu, M. Waqas, F. Huang, G. Abbas and Z. H. Abbas, "A revocable and outsourced multi-authority attribute-based encryption scheme in fog computing," *Computer Networks*, vol. 195, pp. 108196, 2021.
[24] H. Guo, W. Li, M. Nejad and C. C. Shen, "A hybrid blockchain-edge architecture for electronic health record management with attribute-based cryptographic mechanisms," *IEEE Transactions on Network and Service Management*, vol. 1, pp. 1–16, 2022.
[25] X. Qin, Y. Huang, Z. Yang and X. Li, "A blockchain-based access control scheme with multiple attribute authorities for secure cloud data sharing," *Journal of Systems Architecture*, vol. 112, pp. 101854, 2021.
[26] M. Xiao, Q. Huang, Y. Miao, S. Li and W. Susilo, "Blockchain based multi-authority fine-grained access control system with flexible revocation," *IEEE Transactions on Services Computing*, vol. 15, no. 6, pp. 3143–3155, 2021.
[27] J. Yu, S. Liu, M. Xu, H. Guo, F. Zhong *et al.,* "An efficient revocable and searchable MA-ABE scheme with blockchain assistance for C-IoT," *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2754–2766, 2022.
[28] Q. Wu, T. Lai, L. Zhang and F. Rezaeibagha, "Blockchain-enabled multi-authorization and multi-cloud attribute-based keyword search over encrypted data in the cloud," *Journal of Systems Architecture*, vol. 129, pp. 102569, 2022.
[29] Q. Xu, C. Tan, W. Zhu, Y. Xiao, Z. Fan *et al.,* "Decentralized attribute-based conjunctive keyword search scheme with online/offline encryption and outsource decryption for cloud computing," *Future Generation Computer Systems*, vol. 97, pp. 306–326, 2019.
[30] N. Gorasia, R. R. Srikanth, N. Doshi and J. Rupareliya, "Improving security in multi authority attribute based encryption with fast decryption," *Procedia Computer Science*, vol. 79, pp. 632–639, 2016.
[31] Y. Miao, R. H. Deng, X. Liu, K. K. R. Choo, H. Wu *et al.,* "Multi-authority attribute-based keyword search over encrypted cloud data," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 4, pp. 1667–1680, 2019.