electronics

MDPI

Article

A Secure Data-Sharing Scheme for Privacy-Preserving Supporting Node–Edge–Cloud Collaborative Computation

Kaifa Zheng [1,*], Caiyang Ding [2] and Jinchen Wang [3]

[1] School of Cyber Science and Technology, Beihang University, Beijing 100191, China
[2] School of Computer Science and Engineering, Beihang University, Beijing 100191, China
[3] North Information Control Research Academy Group Co., Ltd., Nanjing 211100, China
* Correspondence: zkfbelief@gmail.com

Abstract: The node–edge–cloud collaborative computation paradigm has introduced new security challenges to data sharing. Existing data-sharing schemes suffer from limitations such as low efficiency and inflexibility and are not easily integrated with the node–edge–cloud environment. Additionally, they do not provide hierarchical access control or dynamic changes to access policies for data privacy preservation, leading to a poor user experience and lower security. To address these issues, we propose a data-sharing scheme using attribute-based encryption (ABE) that supports node–edge–cloud collaborative computation (DS-ABE-CC). Our scheme incorporates access policies into ciphertext, achieving fine-grained access control and data privacy preservation. Firstly, considering node–edge–cloud collaborative computation, it outsources the significant computational overhead of data sharing from the owner and user to the edge nodes and the cloud. Secondly, integrating deeply with the "node–edge–cloud" scenario, the key distribution and agreement between all entities embedded in the encryption and decryption process, with a data privacy-preserving mechanism, improve the efficiency and security. Finally, our scheme supports flexible and dynamic access control policies and realizes hierarchical access control, thereby enhancing the user experience of data sharing. The theoretical analysis confirmed the security of our scheme, while the comparison experiments with other schemes demonstrated the practical feasibility and efficiency of our approach in node–edge–cloud collaborative computation.

Keywords: privacy-preserving; node–edge–cloud computation; data sharing; edge computing; attribute-based encryption

Citation: Zheng, K.; Ding, C.; Wang, J. A Secure Data-Sharing Scheme for Privacy-Preserving Supporting Node–Edge–Cloud Collaborative Computation. *Electronics* **2023**, *12*, 2737. https://doi.org/10.3390/electronics12122737

Academic Editors: Di Wu, Jing Li, Xianmin Wang and Mingliang Zhou

Received: 5 May 2023
Revised: 5 June 2023
Accepted: 6 June 2023
Published: 19 June 2023

1. Introduction

The node–edge–cloud collaborative computation paradigm has revolutionized the mode of data sharing, while also introducing a range of privacy or security issues. Considering these, data sharing implementations must prioritize privacy protection through measures such as access control and data encryption. It is imperative to ensure that authorized personnel can access and utilize shared data. Attribute-based encryption is a particularly effective approach for achieving authority control and preserving privacy in node–edge–cloud collaborative computation. In the node–edge–cloud computation scenario, the distributed terminal devices can be considered as "nodes" that generate a significant amount of data. To reduce the burden of data management, data owners tend to outsource the data after encrypting to cloud servers via edge nodes. This centralized approach allows for efficient control of data resources in the cloud, which can be easily accessed and utilized by distributed end-users. However, this approach also means that the data owner loses direct control over the outsourced data, making it difficult to enforce "face-to-face" access control for users. As a result, there is a growing need for fine-grained access control mechanisms and efficient and secure privacy-preserving mechanisms in the node–edge–cloud collaborative computation scenario in the IoT [1,2].

To implement privacy preservation and fined authorization control, relevant researchers have proposed embedding the access policy or authorization identity attribute into the data ciphertext. Shamir and Boneh et al. [3,4] first designed an identity-based encryption access control mechanism, but it was only able to offer coarse-grained authorization control to end-users. To address this, Sahai et al. [5] introduced the concept of attribute-based encryption (ABE) and proposed an access control solution based on user attributes. ABE research primarily revolves around two types of methods: key policy attribute-based encryption (KP-ABE) and ciphertext policy attribute-based encryption (CP-ABE). The linear secret-sharing scheme (LSSS) [6,7] and access control tree [8] are two mainstream basic theories that support CP-ABE schemes. However, the ABE scheme based on the LSSS incurs significant computational overhead in the decryption phase, which is unacceptable for IoT end-users. Therefore, the ABE scheme based on the access control tree is generally preferred for lightweight scenarios.

However, the above-mentioned fined authorization control solution [6–8], which relies on bilinear mapping operations for encryption and decryption, incurs significant computational overhead as the access structure expands. In the node–edge–cloud computation scenario, the lightweight IoT devices with a limited size and resources cannot afford such high-overhead ABE algorithms. Therefore, Zhang et al. [9] gave the outsourcing ABE solution, which transfers the encryption or decryption computing operations to semi-trusted cloud servers or edge nodes and allocates the computational costs to nodes, edges, and the cloud. However, semi-trusted cloud servers or edge nodes may return incorrect or forged decryption results due to curiosity, laziness, or dishonesty. Thus, it is essential to verify the correctness of node–edge–cloud collaborative computation. Some researchers [10–13] proposed an ABE scheme supporting result verification, but it imposes excessive additional costs. However, lightweight IoT devices are usually unwilling to perform excessive calculations to verify the results returned by outsourcing services.

Most existing ABE schemes do not support attribute revocation updates or user revocation, which poses significant security risks and increases the computational cost of the system when applied to the node–edge–cloud collaborative computation scenarios with a large number of end-users. Firstly, among the massive users, there are likely to be some untrusted users that need to be shielded through lightweight attribute revocation and update. Secondly, once some attribute keys are leaked, the system's security cannot be guaranteed. As a result, relevant personnel have given ABE solutions sustaining attribute revocation and update [11,14,15], but these schemes are inefficient and costly when implementing attribute updates.

To achieve data sharing while supporting privacy protection in IoT scenarios, it is necessary to implement a hierarchical access control mechanism. This involves storing data hierarchically based on importance or sensitivity and dividing data users into different levels. For instance, in a medical data scenario, attending doctors, doctors, and nurses can be set as the highest authority, the second-highest authority, and the general authority, respectively, from the highest to the lowest. Patient identity information, disease information, and prescription information can be set as the highest secret, the second-highest secret, and the general secret, respectively, from the highest to the lowest. Higher-level authorized users should be allowed to access lower-level data, while lower-level authorized users should not be able to access higher-level data. Several relevant scholars have proposed hierarchical access control mechanisms [16,17]. However, these solutions are not practical in IoT scenarios due to the lack of user revocation and update, outsourcing computing, and other necessary functions. Therefore, further research is needed to design efficient hierarchical access control mechanisms that are suitable for IoT scenarios. Given the above shortcomings, we designed a data-sharing scheme of attribute-based encryption that supports node–edge–cloud collaborative computation (DS-ABE-CC). First of all, we outsourced a large number of user computational costs to edge nodes and built a low-cost verification outsourcing computing mechanism. Secondly, a low-cost attribute revocation and update mechanism was designed to enhance the security. At the same time, consider-

ing the scenario of multiple owners, users, and edge nodes in the IoT, a key distribution and key agreement mechanism was designed to improve the security of the scheme. Finally, we designed a hierarchical access control scheme to achieve the hierarchical management of authorized users and outsourced data. This paper has the following innovation points and contributions.

Supporting node–edge–cloud computation: The DS-ABE-CC scheme deeply integrates the cloud–edge-end-integrated IoT environment. Then, most of the overhead is transferred to the edge nodes based on edge computing, which realizes node–edge–cloud collaborative computation on the premise of ensuring user privacy and user experience.

Efficient fine-grained access control mechanism: Our scheme is based on the principle of minimizing complex bilinear pairing operations. A new mechanism for key generation, ciphertext generation, and ciphertext decryption has been systematically proposed, which reduces overhead while greatly increasing the computational efficiency.

Key distribution and agreement: Considering the node–edge–cloud scenario with massive terminal access, an innovative shared data model for multiple owners and users was designed, which supports key distribution and key agreement. It makes the scheme have good scenario adaptability and security.

Privacy-preservation for data: Considering the data-privacy-preserving demand, we designed a low overhead hierarchical access control structure of the data to achieve hierarchical access control for users and improve the security and availability of the system. That different authorized users set different level access permissions greatly improves the privacy of the data.

Attribute revocation and update: In the open IoT environment, where there are inevitably malicious users, it is necessary to revoke or update attributes periodically. We designed a lightweight attribute update mechanism to update the attributes and ciphertext, thus improving the security of the DS-ABE-CC scheme.

This paper is organizedd as follows. Section 2 proposes the relevant works. The preliminary knowledge required for this scheme is introduced in Section 3. In Section 4, the overall architecture of the solution is introduced, including the design objectives, system model, and threat model. Section 5 introduces the implementation of the DS-ABE-CC scheme, including the scheme design and function comparison. Section 6 introduces the analysis security and performance. In the end, we give the conclusions.

2. Related Work

In the scenario of node–edge–cloud collaborative computation, a large number of terminal devices generate data that are outsourced to cloud servers via edge nodes, introducing a range of security and privacy. Especially, the shared data are then accessed by terminal users in an "on-demand" manner [18]. For massive users, data owners lose their constraints on users and have to design self-defined access control, to achieve a secure data-sharing solution for privacy preservation. Therefore, some researchers have proposed a fined authorization control scheme, namely ABE, based on a self-defined policy [19]. The CP-ABE scheme is a specific application of the ABE. The main idea is to encrypt the registered attributes into attribute keys, and users who meet the preset attributes can authorize the access. Bethencourt et al. [8] proposed the first access control scheme based on an access control tree and supporting attribute base. First, the access control tree is built according to the self-defined access policy. Secondly, based on the Shamir threshold mechanism, a threshold key distribution and restoration scheme is designed and embedded in the access control tree [8]. Thirdly, in the encryption process, a secret value is inserted into the root node, and the access control tree is established from top to bottom. Finally, in the decryption process, users who meet the predefined attributes can recover the secret value associated with the root node from the bottom up [8]. Subsequently, a large number of scholars have put forward much valuable work based on the access control tree. These works mainly included the ABE scheme for multiple authorized organizations [19,20], supporting the computing outsourcing ABE

scheme [9–13], supporting the user cancellation and update scheme [11,14,15], and supporting the tracking ABE scheme [7].

For the privacy preservation of the data, Brent Waters et al. [6] gave an ABE access control scheme based on a linear secret-sharing scheme (LSSS). This scheme is another way to transform the access control policy into a monotonous Boolean formula, then finally, convert it into an LSSS structure according to the standard formula and embed the difficult problem of cryptography to achieve fine-grained authorization control. Subsequently, a large number of researchers have proposed many creative access control schemes based on the LSSS [6,7]. Cui et al. [21] skillfully combined the linear secret-sharing scheme (LSSS) with ciphertext retrieval to construct a new searchable encryption scheme. Subsequently, Meng et al. [22] and Tseng et al. [23] proposed an improved LSSS-based ciphertext retrieval scheme that supports attribute encryption.

Considering that the fine-grained access control strategy mentioned above is not flexible enough, some scholars [24–27] have given a fined authorization access solution based on the "and" gate. The traditional CP-ABE scheme user terminal bears much computational cost, which makes it difficult to meet the lightweight computing requirements. Green et al. [28] proposed the ABE encryption scheme of end-cloud collaborative computation, which outsourced the user costs to cloud servers to reduce their computational costs, but did not verify the outsourcing results. Considering this, Mao et al. [29] and Zhao et al. [30] gave an attribute-based encryption solution verifying collaborative computation. The length of the ciphertext of this scheme increases with the complexity of the authorization structure. Li et al. [31] proposed a CP-ABE scheme that put key distribution and outsourcing decryption on the cloud, and this scheme supports the verification of the collaborative computation results.

However, the above scheme can only be applied to the scenario where the user's attribute or identity is unchanged. To improve the security, Ostrovsky et al. [32] first put forward the concept of attribute revocation and designed a revocable CP-ABE scheme, adding the symbol "not" to the revoked users to shield them. However, this scheme is inefficient. Meanwhile, Pirretti et al. [33] also proposed a CP-ABE scheme to realize indirect revocation of user attributes. This indirect scheme mainly sets the system parameters, updates the user attribute keys periodically, and then, completes the attribute revocation. The disadvantage is that this mechanism for revoking attributes is inefficient. Later, Xue et al. [11] also proposed some other schemes to support attribute revocation, but these schemes cannot satisfy the demands of efficiency, expressiveness, and security at the same time.

On the other hand, in the privacy-preserving scenario, massive data owners and users are faced with the need to achieve hierarchical data management and hierarchical access control. Wang et al. [17] proposed a hierarchical authorized access scheme for data privacy preservation. First, the hierarchical authorized access model was built. Then, the data and users are grouped and divided into several levels, and the users in different groups are given different access rights. Finally, high-level users can access both the corresponding level of data and lower-level data, but lower-level users are unable to access higher-level data. Liu et al. [34] proposed building a hierarchical authorized access tree and then realized the user-level authorized access mechanism.

However, in the context of node–edge–cloud computation and privacy preservation, new challenges have been brought to data-sharing technology, including attribute revocation and update, collaborative computation, hierarchical access control, and other requirements. The above scheme [11,13–33] cannot implement these mechanisms from multiple dimensions in lightweight mode, namely attribute revocation and update, verifiable collaborative computation, and a hierarchical access system. Considering these problems, we integrated key agreement and edge computing mechanisms and designed a data-sharing scheme of attribute-based encryption that supports node–edge–cloud collaborative computation (DS-ABE-CC), for the IoT scenario.

3. Preliminary

This section will introduce the symbols, formulas, and required theoretical knowledge of this paper.

3.1. Notations

The important symbols and formulas of the paper are as follows:

- Att—The licensing attribute set is $Att = \{att_1, att_2, \cdots, att_u\}$, which is updated periodically.
- S_{DU_j}—$S_{DU_j} \subseteq Att$ represents the attribute set of user DU_j.
- $\Delta_{i,S}(x)$—$\Delta_{i,S}(x) = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$ is the Lagrange coefficient in this scheme.
- Γ—The access control policy is composed of fine-grained access control conditions, such as attribute logic statements linked by "and", "or", and "not".
- $\mathcal{T}$—The access control tree $\mathcal{T}$ is generated according to the access control policy Γ.
- $N(x_1, x_2)$—This represents the x_1-th node in the x_2-th layer from tree $\mathcal{T}$, including the root node, leaf node, transfer node, and inserted control node.
- $index(x)$—$index(x)$ represents the index number corresponding to node x in $\mathcal{T}$. $parent(x)$ is the parent node of node x; $att(x)$ is the attribute of node x; q_x and t_x are the polynomial and threshold values corresponding to node x; d_x is the degree of polynomial q_x.
- $ck = \{ck_1, \cdots, ck_\ell\}$— ck has ℓ content key ck_i, $i \in [1, \ell]$, which corresponds to the ℓ access control level. The ℓ secret values $s'_i = \{s'_1, \cdots, s'_\ell\}$ are embedded in the access control tree $\mathcal{T}$.
- (t, n)—This denotes the threshold corresponding to the nonleaf node. The secret s'_i is decomposed into n shares, where $n > t$. If t shares are collected, the secret s'_i can be restored.
- $\mathcal{D}$—The plaintext document set $\mathcal{D}$ includes N documents, namely $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_N\}$.
- E—The ciphertext document set $E = \{E_1, E_2, \cdots, E_N\}$ includes N documents, which are divided into ℓ sets. $ck = \{ck_1, ck_2, \cdots, ck_\ell\}$ is the document context key set. $E_j = Enc_{ck_i}(\mathcal{D}_j)$, which means encrypt $\mathcal{D}_j$ to E_j by key ck_i.

3.2. Bilinear Mapping

Let G_1 and G_T be two p-order multiplicative cyclic groups, where p is a large prime number. Assume that g is a generator of G_1. We define a mapping $e: G_1 \times G_1 \rightarrow G_T$ with the following properties:

(1) $Bilinear: \forall a, b \in \mathbb{Z}_p^*, e(g^a, g^b) = e(g^b, g^a) = e(g, g)^{ab}$.

(2) $Non\text{–}degenerative: \exists a, b \in \mathbb{Z}_p^*, e(g^a, g^b) \neq 1$.

(3) $Computability: \forall a, b \in \mathbb{Z}_p^*; e(g^a, g^b)$ can be computed efficiently.

3.3. Some Difficult Problems in Cryptography

The discrete logarithm problem (DLP): It refers to the selection of $a \in_R \mathbb{Z}_p^*$, resulting in the tuple (g, g^a) in G_1. The computational complexity of determining the value of a in probabilistic polynomial time (PPT) is very difficult.

The computational Diffie–Hellman problem (CDH): It involves selecting a and b from $\mathbb{Z}_p^*$, which generates the tuple (g, g^a, g^b) in G_1. The computation of g^{ab} in PPT is also highly challenging.

3.4. Decisional Bilinear Diffie–Hellman(DBDH)

Let $e: G_1 \times G_1 \rightarrow G_T$ be a bilinear mapping, where G_1 and G_T are groups, and g is a randomly selected generator from G_1. Consider $a, b, c \in \mathbb{Z}_p^*$, where p is a large prime number. The DBDH hypothesis states that there does not exist an algorithm $\mathcal{B}$ that can distinguish between the tuples $(g, g^a, g^b, b^c, T = e(g, g)^{abc})$ and $(g, g^a, g^b, b^c, T = R_T)$ with non-negligible advantage ϵ, where R_T is a randomly chosen element in G_T. Mathematically, this can be expressed as $|Pr[\mathcal{B}(g, g^a, g^b, b^c, T = e(g, g)^{abc}) = 1] - Pr[\mathcal{B}(g, g^a, g^b, b^c, T =$

$R_T) = 1]| \leq \epsilon$.

3.5. Access Control Structure and Access Control Tree

The identity information of user can be depicted as multiple attributes connected by logical operators "and", "or", and "not", for example "formal staff and (Ph.D. or professional qualification certificate)". Any access policy Γ can be mapped to an access control tree $\mathcal{T}$. The identity attribute set S_{DU_j} can be mapped to the corresponding leaf set.

3.6. Threshold Secret-Sharing Mechanism

A shared secret s_i' is divided into n sub-secrets $s_i' = \{s_{i1}', s_{i2}', \cdots, s_{in}'\}$, and each sub-secret s_{ij}' $(j \in [1, n])$ is distributed to several participants at the same time, which satisfies the following conditions. If there are more than or equal to t sub-secrets s_{ij}', the shared secret s_i' can be recovered. If there are less than t sub-secrets, the shared secret s_i' cannot be recovered. The above is the threshold (t, n) secret-sharing method, the threshold is t, the count of participants is n.

3.7. Kerckhoffs' Principle

The accepted criterion for cryptosystems is Kerckhoffs' principle, which is as follows. The encryption algorithm and design principle of the system are public, and the most-important aspect of a secure system is the key. The security of cryptosystems should depend on the key ck_i, rather than relying on the security of encryption algorithms (DES/AES). Existing papers on searchable encryption or privacy protection all follow this assumption; all focus on the security of the key ck_i when proving security. Similarly, the assumption basis of this article was also this criterion.

4. Construction of DS-ABE-CC

This section describes the overall architecture of DS-ABE-CC. Firstly, we introduce the design goals. Secondly, the paper notation and system model architecture are introduced. Finally, we illustrate the threat model of the system.

4.1. Design Objectives

Considering the IoT environment with mass terminal access, we designed a data-sharing scheme of attribute-based encryption that supports node–edge–cloud collaborative computation (DS-ABE-CC). The design goals were as follows.

Node–edge–cloud collaborative computing: Outsourcing the data owner's and user's computing to edge nodes, our scheme performs node–edge–cloud collaborative computation, reduces user computing overhead, and verifies the correctness of the outsourcing computing results.

Support diversified interactive data: The interactive data of the terminal cover various kinds of data including images, text, audio, video, etc. We used structured data documents to store and describe the data resources with fine-grained features.

Privacy preservation for shared data: The DO uses symmetric encryption to encrypt data documents $\mathcal{D}_i$, namely $E_i = Enc_{ck_i}(\mathcal{D}_i)$, while the DU uses symmetric keys to decrypt the data, $\mathcal{D}_i = Dec_{ck_i}(E_i)$. The encryption key ck_i is embedded in the access policy's ciphertext to ensure the confidentiality, privacy, and availability of the data file.

A lightweight hierarchical CP-ABE mechanism: We divided the massive data users into several levels; meanwhile, the data were also divided into several security levels. A low overhead hierarchical access control method is proposed and constructed,which can support hierarchical access control for end-users.

Support attribute and ciphertext updates: Considering that users are dynamically updated in the Internet of Things, a mechanism supporting attribute revocation and update was designed, to achieve attribute and ciphertext updates.

4.2. Overall Model

4.2.1. Overall Architecture

Considering the background of the node–edge–cloud and privacy-preserving in the IoT, we designed a data-sharing scheme of attribute-based encryption that supports node–edge–cloud collaborative computation (DS-ABE-CC). This system consists of five types of entities, including the key generation center (KGC), cloud service provider (CSP), edge nodes (ENs), data owners (DOs), and end-users (DUs). As shown in Figure 1, we regarded the terminal that generates the data as the data owner, and the generated data are outsourced to the cloud server through the edge node. Massive end-users are regarded as data users, who access the cloud service system through edge nodes. The model is shown in Figure 1. The DOs and DUs communicate with the CSP through the ENs by an open channel. The specific interaction process is as follows:

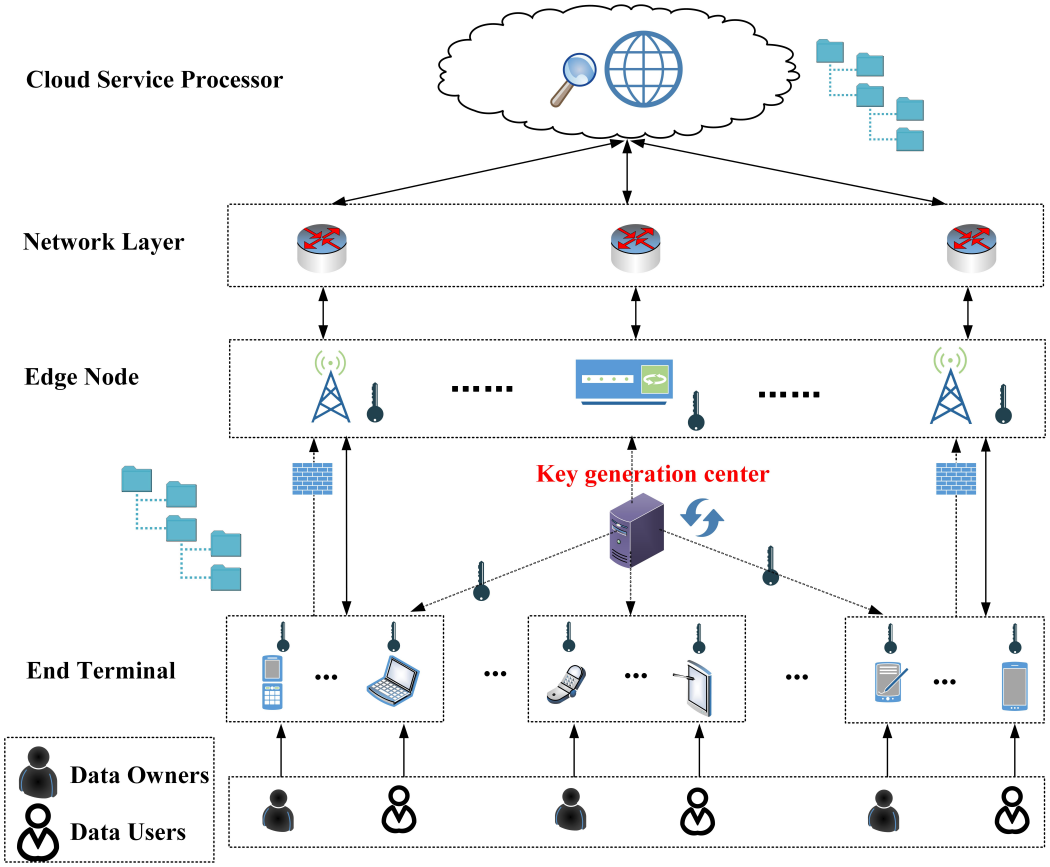


Figure 1. Overall architecture.

1. **Key generation center (KGC):** This entity is completely trusted and referred to as the KGC. The KGC is responsible for managing the system's public parameter PK and the master key MK. It facilitates key distribution and attribute management for all entities within the system.
2. **Cloud service provider (CSP):** This entity is a semi-trusted entity that provides ciphertext storage and computing services. Firstly, it can faithfully execute the preset calculation protocol and return the correct results. Secondly, it curiously guesses the privacy of each entity and tries to crack the encrypted ciphertext.
3. **Data owners (DOs):** The DOs are responsible for defining the access structure and performing data encryption operations, embedding the access control policy Γ into the encrypted file E with the help of ENs, then delegating it to the CSP.
4. **Data users (DUs):** These entities generally refer to the end-users of the IoT, who access the data stored in the CSP. DUs who meet predefined access control policies can access, download, and decrypt target ciphertext.

5. **Edge nodes (ENs):** The ENs are entities between the DOs, DUs, and CSP centers, which provide certain storage, computing, and other resources. In this model, part of the computing overhead of the DOs and DUs is transferred to the edge nodes (ENs) to reduce the users' computing overhead. The ENs is considered "honest and curious" like CSP.

4.2.2. Overview of DS-ABE-CC

The DS-ABE-CC scheme consists of five main algorithms, including initialization ($Setup$), key generation and distribution ($KeyGen$), encryption ($Encrypt$), ciphertext decryption ($Decrypt$), and attribute update ($Update$).

- $\mathbf{Setup}(\mathbf{1}^{-}) \rightarrow (\mathbf{MK}, \mathbf{PK})$: The input includes the initial security parameter μ, and the output parameters from the KGC consist of the public parameter PK and the master key MK for the system.
- $\mathbf{KeyGen}(\mathbf{MK}, \mathbf{S_{DU_j}}) \rightarrow ((\mathbf{sk}, \mathbf{pk}), \mathbf{SK})$: The KGC distributes keys for each entity. The algorithm is executed by the KGC, which inputs the user DU_j, attribute set S_{DU_j}, and master key MK, then generates the corresponding attribute key SK for each DU_j.
- $\mathbf{Encrypt}(\mathbf{PK}, \mathbf{\Gamma}, \mathbf{ck}) \rightarrow \mathbf{CT}$: The data owner DO_i and the edge nodes (ENs) executed this algorithm in cooperation. The public parameter PK, access control policy Γ, and content key ck are input. According to Γ, the plaintext ck is encrypted into the ciphertext CT.
- $\mathbf{Decrypt}(\mathbf{PK}, \mathbf{SK}, \mathbf{CT}) \rightarrow \mathcal{D}$: The algorithm is executed by user DU_j and edge node EN in cooperation, which input the parameters PK, ciphertext CT, and attribute private key SK of user DU_j. If user DU_j can meet the preset access control policy Γ, then the ciphertext CT can be decrypted to plaintext content key ck_i. Otherwise, the plaintext content key ck_i cannot be returned.
- $\mathbf{Update}(\mathbf{S_x}, \mathbf{PK}, \mathbf{SK}, \mathbf{CT}) \rightarrow (\mathbf{SK}', \mathbf{CT}')$: Input the attribute S_x to be updated, public key PK, ciphertext CT, and user's attribute key SK. Finally, the attributes key SK and ciphertext CT will be updated as SK' and CT', separately.

4.3. Threat Model

This section introduces the threat model of the DS-ABE-CC system in the IoT environment. Based on the information possessed by the CSP and DUs, we adopted the following threat models [11]:

Choose plaintext attack (CPA): An adversary Adv can arbitrarily choose plaintexts and their corresponding ciphertext, then try to deduce the association between their ciphertext and plaintexts or crack the indistinguishability of two plaintext ciphertexts.

Definition 1. *With the DBDH problem, the DS-ABE-CC system can achieve CPA security if there is no Adv and can win the below game in probabilistic polynomial time (PPT) with a non-negligible advantage* ϵ.

Setup: The challenger *Chal* runs an algorithm $Setup(1^{\mu})$ to generate parameters, keep the master key MK, and send the PK to the Adv.

Phase 1: The Adv selects any attribute set $S = \{S_1, \cdots, S_{u'}\}$, which does not meet the access control structure $\mathcal{T}^*$. The *Chal* runs the algorithm *KeyGen* to generate the corresponding attribute ciphertext SK and returns it to the Adv.

Challenge: The Adv selects two plaintexts M_0, M_1 $(M_0 \neq M_1)$, which will be sent to the *Chal*. The *Chal* randomly selects $\mathfrak{b} \in \{0, 1\}$ and generates $CT^* = Encrypt(PK, M_{\mathfrak{b}}, \mathcal{T}^*)$ based on $M_{\mathfrak{b}}$. Then, the *Chal* returns CT^* to the Adv.

Phase 2: Similar to Phase 1, the Adv will inquire in PPT. The limitation is that the inquiries in Phase 1 or the M_0, M_1 cannot be repeated.

Guess: The Adv outputs $\mathfrak{b}'$. If $\mathfrak{b}' = \mathfrak{b}$, the Adv wins the game. Otherwise, the Adv loses the game. The probability of Adv winning this safety game is $|Pr(\mathfrak{b}' = \mathfrak{b}) - \frac{1}{2}|$.

5. Realization of DS-ABE-CC Scheme

DS-ABE-CC is oriented toward multiple DOs and DUs in node–edge–cloud collaborative computation IoT scenarios. To facilitate the description, a data owner DO_i and data user DU_j are used as an example to describe the implementation principle of DS-ABE-CC.

5.1. Hierarchical Access Control Tree Supporting Privacy Preservation

The DOs divide all documents $\mathcal{D}$ into ℓ document sets according to their importance and privacy and then design ℓ levels of access rights in turn. Namely, the first set has the highest access rights, the second set the second level, and the ℓ-th set the lowest level of access rights. As shown in Figure 2, some control nodes are inserted to establish a hierarchical access control tree [17,34], where $N1$ and $N2$ are null attributes inserted in the control nodes of the tree $\mathcal{T}$. All nodes in the tree $\mathcal{T}$ can be divided into four types as follows:

- **Transfer node** (**TN**): The sub-nodes of node x contain at least one threshold, and node x is defined as the transfer node. $TN{-}CT(x)$ represents the threshold sub-node set of transfer node x, namely $TN{-}CT(x) = \{ch_1, ch_2, \cdots, ch_j\}$ [17,34]. The node ABCDE is the transfer node.
- **Control node** (**CN**): The parent node of a null attribute is defined as the control node. The BD in tree $\mathcal{T}$ represents the control node.
- **Hierarchical node** (**TN**)-**CN**: The transfer node TN set minus the control node CN set is the hierarchical node. For example, the node ACE is the hierarchical node.
- **Leaf node** (**LN**): The leaf nodes are the lowest nodes in the access control tree $\mathcal{T}$, in which each leaf represents an attribute att_i.

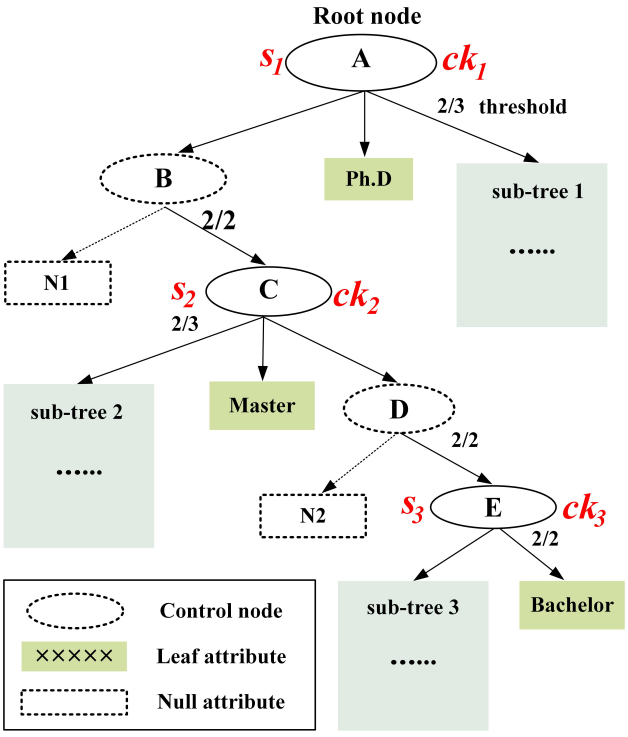


Figure 2. Hierarchical access control tree. The nodes ABCDE are the transfer node, the BD are the control node, ACE are the hierarchical node, and N1 N2 are the null node.

We established a three-level access control tree, namely $\ell = 3$, as shown in Figure 2 The constructed access control tree meets the following properties:

- If the content key ck_1 associated with the root node can be recovered by the DUs, then they have the highest level of access rights.

- If the content key ck_2 can be recovered by the DUs, then they have the second-level access right.
- If the content key ck_3 can be recovered by the DUs, then they have the third level of access.

The document set $\mathcal{D}$ is encrypted to E by symmetric key ck, which is finally sent and stored in the CSP. ck_1 is the highest level, and ck_3 is the lowest level.

Claim 1: If the users can decrypt ck_i, then they are able to decrypt $ck_{i+1}, \cdots, ck_\ell$. However, when the users can decrypt ck_{i+1}, they cannot decrypt $ck_1, \cdots, ck_i$.

5.2. The Secure Data Sharing Scheme Supporting Node–Edge–Cloud Computation

The data-sharing scheme of attribute-based encryption that supports the node–edge–cloud collaborative computation (DS-ABE-CC) proposed in this paper consists of five algorithms, including initialization (*Setup*), encryption (*Encrypt*), key generation and distribution (*KeyGen*), ciphertext decryption (*Decrypt*), and attribute update (*Update*). The specific algorithms are described as follows in Figure 3.

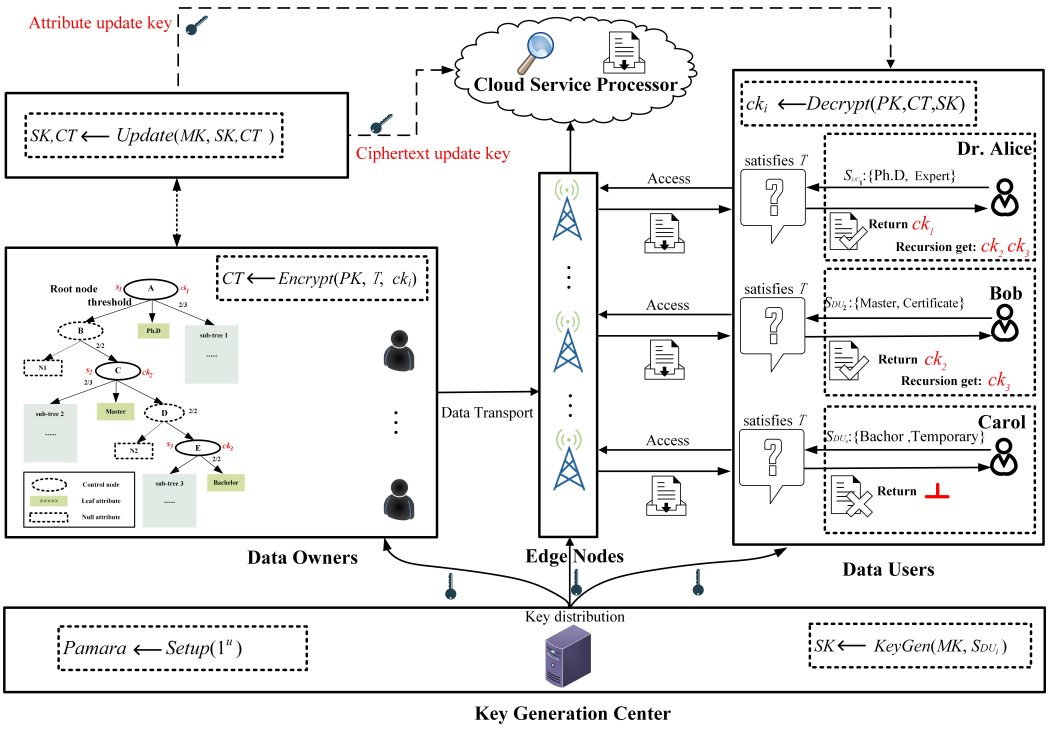


Figure 3. Example diagram of the DS-ABE-CC scheme. The five algorithms includes initialization (*Setup*), encryption (*Encrypt*), key generation and distribution (*KeyGen*),ciphertext decryption (*Decrypt*) and attribute update (*Update*).

1. $\mathbf{Set}(\bar{\mathbf{1}}) \rightarrow (\mathbf{MK}, \mathbf{PK})$: G_1 is a p-order multiplicative cyclic group, in which g is a generator. The hash function is defined as $H : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$. Randomly selecting two parameters $\alpha, \beta \in_R \mathbb{Z}_p^*$, the KGC generates master key MK and public parameter PK.

$$\begin{aligned} MK &= \{\alpha, \beta\}; \\ PK &= \{H, G_1, g, g^\beta, e(g,g)^\alpha, e(g,g)^{\alpha\beta}\} \end{aligned} \tag{1}$$

2. $\mathbf{KeyGen}(\mathbf{MK}, \mathbf{S_{DU_j}}) \rightarrow ((\mathbf{sk}, \mathbf{pk}), \mathbf{SK})$: First, the KGC distributes keys for each entity. Then, input the master key MK and attribute set of user S_{DU_j}, and the attribute key SK is generated for each end-user DU_j.
   - **Key distribution** : Firstly, the KGC generates a secret key $sk_{DO_i} = \gamma \in_R \mathbb{Z}_p^*$ for each data owner DO_i, whose public key is $pk_{DO_i} = g^\gamma$. In the same manner,

the KGC generates a secret key $sk_{DU_j} = r \in_R \mathbb{Z}_p^*$ for each end-user DU_j, whose public key is $pk_{DU_j} = g^{\beta r}$, and secret key $sk_{EN_k} = e \in_R \mathbb{Z}_p^*$ for each edge node EN_k, whose public key is $pk_{EN_k} = g^e$. The public key of each entity is broadcast in public.

- **Key agreement** : Firstly, the KGC issues the identity shield token $Token = g^{\alpha\beta/r} g^{\alpha+\beta}$ and the attribute set $S_{DU_j} = \{S_1, \cdots, S_{u'}\} \subseteq Att$ to the end-user DU_j. From the public key $pk_{EN_k} = g^e$, the primitive attribute key of the user DU_j is SK after key agreement as follows.

$$SK = \{D = g^{\alpha\beta} g^{(\alpha+\beta)r} g^{r^2 e}, D' = e(g,g)^{-\beta r^2}\} \tag{2}$$

  Finally, the user DU_j sends primitive attribute key SK with its own attribute S_{DU_j}, namely $(S_{DU_j}||SK)$, to the edge node EN_k.
- **For edge node EN_k**. Finally, according to attribute set S_{DU_j} of user DU_j, the edge node EN_k generates a random value r_i for every target attribute $j \in S_{DU_j}$. Then, the attribute key of user DU_j is SK' as follows.

$$SK' = \{\forall j \in S,\ D_j = g^{r_j + H(j) + r}, D'_j = g^{r_j}\} \tag{3}$$

Finally, $SK||SK'$ is sent to the CSP for data sharing by edge node EN_k.

3. **Encrypt**$(\mathbf{PK}, \mathbf{\Gamma}, \mathbf{ck}) \rightarrow \mathbf{CT}$: This inputs access control policy Γ, public key PK, and context key $ck = \{ck_1, \cdots, ck_\ell\}$. The owners and upstream ENs cooperate to complete encryption. First, the owner only needs to perform lightweight constant operations to generate pre-encrypted ciphertext, which is sent to the upstream edge node. Then, the EN will complete the remaining encryption calculations. The ciphertext CT will be generated. Assume the private and public key pair of the upstream edge node of DO_i is $(e', g^{e'})$.
   - **Encryption in data owner $\mathbf{DO_i}$**: The DO_i defines the access level of each document $D_j (j \in [1, N])$ and key ck_i $(i \in [1, \ell])$, then encrypts D_j into ciphertext $E_j (j \in [1, N])$ based on the corresponding level key ck_i $(i \in [1, \ell])$, namely $E_j = Enc_{ck_i}(\mathcal{D}_j)$, $j \in [1, N]$. It will obtain a set of dictionaries containing encrypted documents and corresponding levels, namely $\{E_j : i\}$, $i \in [1, \ell]$, $j \in [1, N]$. Lastly, the DO_i randomly selects $s_i = \gamma^i \in_R \mathbb{Z}_p^*, (\forall i \in [1, \ell])$ and encrypts the content key $ck = \{ck_1, \cdots, ck_\ell\}$ into ciphertext CT.

$$\begin{aligned} CT = \{&\Gamma, \{E_j : i\}, \widetilde{C} = ck_i e(g,g)^{\alpha\beta s_i}, \xi = g^{H(ck_i)}, \\ &C' = g^{s_i}, C'' = e(g,g)^{\alpha s_i} | i \in [1, \ell], j \in [1, N]\} \end{aligned} \tag{4}$$

     Then, the CT is sent to edge node EN_k.
   - **Encryption in edge node $\mathbf{EN_k}$**: First, the EN_k randomly selects $s'_i = (e')^i \in_R \mathbb{Z}_p^*$, $(\forall i \in [1, \ell])$, to obtain $s' = \{s'_1, \cdots, s'_\ell\}$. Then, for every nonleaf node x in $\mathcal{T}$ from top to bottom, sequentially, a d_x-order constant polynomial q_x $d_x = t_x - 1$ is generated, where t_x is the threshold value. Secondly, it sets $q_R(0) = s'_1$ in the root node. Subsequently, following the same rule, the remaining child node x of the root node is set with the corresponding constant polynomial q_x. If node x is a hierarchy node, then $q_x(0) = q_{\ell_i}(0) = s'_i$. If node x is not a hierarchy node, then $q_x(0) = q_{parent}(index(x))$. Lastly, $s'_1, \cdots, s'_\ell$ must be embedded in the root node and $\ell - 1$ hierarchical nodes in the tree $\mathcal{T}$ in turn. An access control tree $\mathcal{T}$ is built for access policy Γ. Each leaf node in $\mathcal{T}$ corresponds to a licensing attribute. Finally, $Att = \{att_1, att_2, \cdots, att_u\}$ is the licensing attribute set. The X in tree T is a transport node set. $TN-CT(x_1, x_2)$ is the children of the threshold set of transport node $(x \in X)$, namely $TN-CT(x_1, x_2) = \{ch_1, \cdots, ch_j, \cdots\}$. The data

owner computes $\widehat{C_{x,j}}$ for each node for set X and for all $j = 1, 2, \cdots$ as follows in Equation (5).

$$\widehat{C_{x,j}} = g^{q_{ch_j}(0)+s'_i} \tag{5}$$

The EN_k extracts all leaf nodes y of access control tree $\mathcal{T}$, to build leaf node set Y. A random number $r_y \in \mathbb{Z}_p^*$ is generated for each leaf . Based on access control tree $\mathcal{T}$, the content key will be computed as the final ciphertext CT.

$$\begin{aligned} CT = &\{\mathcal{T}, \{E_j : i\}, \widetilde{C} = ck_i e(g,g)^{\alpha\beta s_i}, \xi = g^{H(ck_i)}, \\ &C = g^{s_i}, C' = g^{s_i - s'_i}, C'' = e(g,g)^{\alpha s_i} | i \in [1,\ell], j \in [1,N] \\ &\{\forall y \in Y, C_y = g^{q_y + r_y}, C'_y = g^{H(att(y))+r_y}\} \\ &\{\forall x \in X, \widehat{C_{x,j}}\}\} \end{aligned} \tag{6}$$

The final ciphertext CT encrypted by the edge node EN_k is sent to the CSP.

4. **Decrypt**$(\mathbf{CT}, \mathbf{SK}, \mathbf{PK}) \rightarrow \mathcal{D}$: Input the ciphertext CT, user's attribute key SK, and public key PK. If the user attributes S_{DU_j} can meet the preset access policy, ck_i can be decrypted from CT. Otherwise, the corresponding ck_i cannot be decrypted, and $\perp$ will be output. The edge node EN_k first decrypts the ciphertext and sends the semi-decrypted ciphertext to the user DU_j. Then, the DU_j only needs to execute a lightweight constant calculation to decrypt the ciphertext. Meanwhile, the users DU_j can verify the decrypted results.

- **Decryption in the edge nodes $\mathbf{EN_k}$**: In the access control phase, the following protocol is implemented for attribute verification. After receiving the SK, the EN_k will map the attributes $S_{DU_j} = \{S_1, \cdots, S_{u'}\} \subseteq Att$ to $\{leaf_i\}$. We define the recursive operation $DR(CT, SK, x)$, in which x represents a node in $\mathcal{T}$. If x belongs to the leaf node, namely $x \in LN$, set $i = attr(x)$, then the recursive operation is executed as Equation (7).

$$DR(CT, SK, x) = \begin{cases} e(\frac{C_x \cdot D_i}{C'_x \cdot D'_i}, pk_{DU_j}) \cdot D', & (i \in S) \\ \perp, & (i \notin S) \end{cases} \tag{7}$$

Let $R_x = e(\frac{C_x \cdot D_i}{C'_x \cdot D'_i}, pk_{DU_j}) \cdot D'$, which can be computed as follows in Equation (8).

$$\begin{aligned} R_x &= e(\frac{C_x \cdot D_i}{C'_x \cdot D'_i}, pk_{DU_j}) \cdot D' \\ &= e(\frac{g^{q_y} g^{(r_i + H(j) + r)}}{g^{H(att(y))} g^{r_i}}, g^{\beta r}) \cdot D' \\ &= e(g,g)^{\beta r q_y} \end{aligned} \tag{8}$$

When x is a non-leaf node in tree $\mathcal{T}$, the operation $DR(CT, SK, z)$ is required for each sub-node z of x. Assume there is a random node set S_x, whose size is k_x. The recursive process continues if all child nodes of x are in set S_x. Otherwise, $F_z = \perp$ will be obtained. Let $i = index(z)$, $S'_x = \{index(z) : z \in S_x\}$. We obtain the below Equation (9).

$$\begin{aligned} F_x &= \prod_{z \in S_x} F_z^{\Delta_{i,S_x}(0)} \\ &= \prod_{z \in S_x} (e(g,g)^{r\beta q_z(0)})^{\Delta_{i,S_x}(0)} \\ &= e(g,g)^{\beta r q_x(0)} \end{aligned} \tag{9}$$

The $e(g,g)^{r\beta s'_i}$ can be obtained when the attribute set S_{DU_j} can meet the *i-th*-level access control condition. Meanwhile, the lower-level correlation value $e(g,g)^{r\beta s'_j}$ $(j \geq i)$ can be obtained through recursive calculation. The specific recursive operation is as follows in Equation (10). Through analysis, our scheme has a lower overhead to achieve hierarchical access control than other schemes [17,34].

$$\begin{aligned} F_{i+1,j} &= \frac{e(\widehat{C_{x,j}}, pk_{DU_j})}{F_i} \\ &= \frac{e(\widehat{C_{x,j}}, g^{\beta r})}{F_i} \\ &= e(g,g)^{r\beta s'_i} \end{aligned} \tag{10}$$

Therefore, $e(g,g)^{r\beta s'_i}, \cdots, e(g,g)^{r\beta s'_\ell}$ can be calculated in turn. Meanwhile, when the attribute set S_{DU_j} can meet the authorization conditions Γ, the following calculation can be performed correctly as follows in Equation (11).

$$\begin{aligned} A &= DR(CT, SK, x_{l_i}) = e(g,g)^{\beta r s'_i} \\ Z &= \frac{\widetilde{C} \cdot A \cdot e(C', pk_{DU_j})}{e(C,D)} \\ Z &= \frac{ck_i e(g,g)^{\alpha\beta s_i} \cdot A \cdot e(g^{\beta(s_i - s'_i)}, g^r)}{e(g^{s_i}, g^{\alpha\beta} g^{r(\alpha+\beta)} g^{r^2 e})} \\ Z &= ck_i e(g,g)^{-r^2 e s_i} e(g,g)^{-\alpha r s_i} \end{aligned} \tag{11}$$

Finally, the edge node EN_k sends $Z||e(g^{s_i}, g^e)||e(g,g)^{\alpha s_i}$ to the end-user DU.

- **Decryption in the end user $\mathrm{DU_j}$**: Finally, end-user DU_j can compute and restore the context key ck_i through a simple exponential operation based on sk_{DU_j} as follows in Equation (12).

$$\begin{aligned} ck'_i &= ck_i e(g,g)^{-r^2 e s_i} e(g,g)^{-\alpha r s_i} e(g^{s_i}, pk_{EN_k})^{r^2} e(g,g)^{\alpha s_i r} \\ &= ck_i e(g,g)^{-r^2 e s_i} e(g,g)^{-\alpha r s_i} e(g^{s_i}, g^e)^{r^2} e(g,g)^{\alpha s_i r} \end{aligned} \tag{12}$$

- **Verification in data user $\mathrm{DU_j}$**: In the IoT scenario, the users need to verify whether their decrypted ck'_i is correct or not. The user checks whether the following equation is true as follows.

$$\xi =^? g^{H(ck'_i)} \tag{13}$$

If Equation (13) holds, then the outsourcing computing is correct, and the user DU_j can obtain corresponding access level i' and decrypt the plaintext document using the symmetric key ck'_i. Otherwise, there are some errors in the outsourcing computing, which means that the legitimate user has obtained the wrong decryption result, then "$\perp$" will be obtained.

- **User–edge–cloud collaborative computing**: The CSP downloads the ciphertext set E' of the corresponding level ck'_i to users DU_j through edge nodes EN_k, and the user DU_j can decrypt the plaintext document using the symmetric key $\mathcal{D}_j = Dec_{ck'_i}(E_j)$ as follows.

$$\mathcal{D}_j = Enc_{ck_i}(E_j), j \in [1, N'] \tag{14}$$

This scheme is based on lightweight methods to achieve decryption and verification.

5. **Update**$(\mathbf{S_x}, \mathbf{PK}, \mathbf{SK}, \mathbf{CT}) \rightarrow \{\mathbf{SK}', \mathbf{CT}'\}$**:** scenarios in the IoT, the authorization attribute is dynamic. The malicious DUs can use outdated attribute keys to access secret data. Considering these, our scheme needs to support attribute update [35]. Input the attribute S_x to be updated, public key PK, ciphertext CT, and the user's attribute key SK. Generate attribute update key t_1 and ciphertext update key t_2. Then, the attribute key SK and ciphertext CT will be updated.
   - **Attribute update:** Assume that the attribute $S_x \in Att$ is revoked or updated to S_x^*. The KGC generates attribute update key $t_1 = g^{(H(S_x^*)-H(S_x))}$, then sends the attribute update secret t_1 to the DUs that hold these attributes, but have not been revoked. Finally, the DUs will update its attribute key as follows in Equation (15).

$$\forall S_x \rightarrow S_x^*, D_y' = t_1 \cdot g^{H(S_x)+r+r_j} = g^{H(S_x^*)+r+r_j} \tag{15}$$

     At the same time, the KGC sends the attribute update key t_1 to the CSP, which updates the ciphertext CT as follows in Equation (16).

$$\forall S_x \rightarrow S_x^*, C_y' = t_1 \cdot g^{H(S_x)} = g^{H(S_x^*)} \tag{16}$$

   - **Ciphertext re–encryption:** When the data owner DO_i updates a secret value s_i to s_i^*, the DO generates ciphertext update key $t_2 = (t_{21}, t_{22}) = (e(g,g)^{\alpha\beta s_i^* - s_i}, g^{s_i^* - s_i})$ and sends it to the CSP. The CSP can refresh the ciphertext as follows in Equation (17).

$$\begin{aligned} \widetilde{C} &= ck_i e(g,g)^{\alpha\beta s_i} t_{21} \\ &= ck_i e(g,g)^{\alpha\beta s_i} e(g,g)^{\alpha\beta(s_i^* - s_i)} \\ &= ck_i e(g,g)^{\alpha\beta s_i^*} \\ C &= g^{s_i} t_{22} = g^{s_i^*}, \\ C' &= g^{s_i - s_i'} t_{22} = g^{s_i^* - s_i'} \end{aligned} \tag{17}$$

     Finally, the attribute revocation or update and ciphertext update are realized through these lightweight methods.

5.3. Feasibility Verification

This scheme can achieve hierarchical encryption and decryption, and has good feasibility.

Theorem 1. *If the users can decrypt ck_i, then they are able to decrypt $ck_{i+1}, \cdots, ck_\ell$. However, when users can decrypt ck_{i+1}, they cannot decrypt $ck_1, \cdots, ck_i$.*

Proof of Theorem 1. We assumed that the attribute set S_{DU_j} can meet the *i-th* level access control condition, then $e(g,g)^{r\beta s_i'}$ will be obtained. This $e(g,g)^{r\beta s_i'}$ is the core of decrypting the content key ck_i. Based on the collaborative calculation between the EN_k Equation (11) and the data user DU_j Equation (12), ck_i will be decrypted.

Meanwhile, the correlation value $e(g,g)^{r\beta s_{i+1}'}, \cdots, e(g,g)^{r\beta s_\ell'}$ of a lower level can be obtained through recursive calculation. The specific recursive operation is as follows in Equations (18) and (19).

$$\begin{aligned} F_{i+1,j} &= \frac{e(\widehat{C_{x,j}}, pk_{DU_j})}{F_i} \\ &= \cdots \\ &= e(g,g)^{r\beta q_{ch_j}(0)}, \ (ch_1(0), ch_2(0), \cdots) \end{aligned} \tag{18}$$

Then, we choose the correct child node value $\widehat{C_{x,j}}'$ of $ch_j(0)$ and perform recursive calculations.

$$\begin{aligned} F_{i+2,j} &= \frac{e(\widehat{C_{x,j}}', pk_{DU_j})}{F_{i+1,j}} \\ &= \cdots \\ &= e(g,g)^{r\beta s'_{i+1}} \end{aligned} \tag{19}$$

One recursive operation can solve for the $e(g,g)^{r\beta s'_{i+1}}$ of the sub-level, based on $e(g,g)^{r\beta s'_i}$. Therefore, $e(g,g)^{r\beta s'_{i+1}}, \cdots, e(g,g)^{r\beta s'_\ell}$ can be calculated in turn. Namely, the content key $ck_i, \cdots, ck_\ell$ can be decrypted.

On the contrary, the null attributes are inserted into one child node of each control node, and the threshold for each control node is "2/2", making it impossible to calculate $e(g,g)^{r\beta s'_i}$ based on $e(g,g)^{r\beta s'_{i+1}}$. For the same reason, $ck_1, \cdots, ck_{i-1}$ is even less likely to be solved. Namely, if users can decrypt ck_{i+1}, they cannot decrypt $ck_1, \cdots, ck_i$.

According to the above Equation (19), we can easily obtain the i-level content key ck_i. The corresponding document can be decrypted, namely $\mathcal{D}_j = Dec_{ck_i}(E_j)$. □

5.4. Functional Comparison

As shown in Table 1, compared with other mainstream schemes [6,8,12,13,15,17,34,35], our DS-ABE-CC scheme embeds key distribution and agreement mechanisms, which is more suitable for the massive terminal access and lightweight scenarios in the IoT and has more complete functions. The BSW scheme [8] and Water's scheme [6] are two kinds of classic fine-grained access schemes. Zhang's scheme [12] and Li's scheme [13] are verifiable outsourcing computing ABE schemes. Wang's scheme [17] and Liu's scheme [34] are advanced hierarchical access control schemes. Dong's scheme [15] and Miao's scheme [36] are advanced schemes that support attribute updates and revocation.

Table 1. Function comparison.

Scheme	F1	F2	F3	F4	F5	F6
Our scheme	✓	✓	✓	✓	✓	✓
BSW Scheme [8]	✓	–	–	–	–	–
Water's scheme [6]	✓	–	–	–	–	–
Zhang's scheme [12]	✓	✓	✓	–	–	–
Li's Scheme [13]	✓	✓	✓	–	–	–
Wang's Scheme [17]	✓	–	–	–	–	✓
Liu's Scheme [34]	✓	–	–	–	–	✓
Dong's Scheme [15]	✓	–	–	✓	–	–
Miao's Scheme [36]	✓	✓	–	✓	–	–

"F1"denotes fine-grained access control, "F2"denotes outsourcing encryption or decryption, "F3"denotes outsourcing calculation verification, "F4"denotes attribute update and revocation, "F5"denotes key distribution and agreement, "F6"denotes hierarchical access control.

6. Security and Performance Analysis

6.1. Privacy and Security Analysis

6.1.1. Privacy Analysis

The ciphertext of the DS-ABE-CC scheme has good privacy, and the attribute ciphertext is unlinkable.

Theorem 2. *The ciphertext has good privacy. The ENs and the CSP cannot crack any plaintext information from the ciphertext CT. Meanwhile, even the identical attribute $j \in Att$ will generate different attribute ciphertexts SK for different users.*

Proof of Theorem 2. We, respectively, describe the privacy security of the ciphertext CT and the unlinkability of the attribute ciphertext SK:

(1) Privacy of the ciphertext CT: The ENs and CSP cannot crack the secret s_i, ck_i, or $att(x)$ from the ciphertext CT.

- **For the ENs**: The ciphertext CT observed by the edge node is as follows in Equation (20).

$$\begin{aligned} CT = &\{\Gamma, \widetilde{C} = ck_i e(g,g)^{\alpha\beta s_i}, \xi = g^{H(ck_i)}, \\ &C' = g^{s_i}, C'' = e(g,g)^{\alpha s_i} | i \in [1,\ell]\} \end{aligned} \tag{20}$$

  The ENs cannot obtain the value $e(g,g)^{\alpha\beta s_i}$ from g^{α}, g^{β} and g^{s_i} based on the DBDH problem. Furthermore, ck_i cannot be obtained.
- **For the CSP:** The ciphertext CT observed by the CSP is as follows in Equation (21).

$$\begin{aligned} CT = &\{\mathcal{T}, \widetilde{C} = ck_i e(g,g)^{\alpha\beta s_i}, \xi = g^{H(ck_i)}, \\ &C = g^{s_i}, C' = g^{s_i - s_i'}, C'' = e(g,g)^{\alpha s_i} | i \in [1,\ell] \\ &\{\forall y \in Y, C_y = g^{q_y + r_y}, C_y' = g^{H(att(y)) + r_y}\} \\ &\{\forall x \in X, \widehat{C_{x,j}} = g^{(q_{ch_j}(0) + s_i')}\}\} \end{aligned} \tag{21}$$

  In the same way, the CSP cannot obtain the secret ck_i or s_i from the ciphertext CT. From the DL problem, the value $H(att(y))$ cannot be computed from C_y or C_y'.

(2) Attribute ciphertext unlinkability: Even the same attribute $j \in Att$ will randomly generate different attribute ciphertexts as follows in Equation (22).

$$\begin{aligned} SK &= \{D = g^{\alpha\beta} g^{(\alpha+\beta)r} g^{r^2 e}, D' = e(g,g)^{-\beta r^2}\} \\ SK' &= \{\forall j \in S, D_j = g^{r_j + H(j) + r}, D_j' = g^{r_j}\} \end{aligned} \tag{22}$$

From the DLP, it is difficult to solve $H(j)$ and r_j. Meanwhile, $\frac{D_j}{D_j'} = g^{(r+H(j))}$, the algorithm $KeyGen(MK, S_{DU_j})$ is executed each time, which inputs the secret key r of different users and the random number r_j, and even the same attribute $j \in Att$ will generate different D_j and D_j'. This means that the value $\frac{D_j}{D_j'}$ is different, which means attribute ciphertexts are unlinkable. □

6.1.2. Security Analysis

The scheme can resist the chosen-plaintext attack (CPA). Because $E = Enc_{ck_i}(\mathcal{D})$, the context key ck_i is only shared between DO_i and authorized DU_j. The security of the document set E is guaranteed by key ck_i, based on the accepted criterion of Kerckhoffs' principle. We only need to analyze the security of the ciphertext CT.

Theorem 3. *If there is an adversary Adv that can destroy the DS-ABE-CC system in the chosen-plaintext attack (CPA) with a non-negligible advantage ϵ in PPT, then we construct a simulator $\mathcal{B}$ to solve the DBDH problem with the advantage $\epsilon/2$.*

Proof of Theorem 3. Given a security parameter μ, the challenger *Chal* chooses $a, b, c \in_R \mathbb{Z}_p^*$, a generator $g \in G_1$, and $R_T \in_R G_T$ that is a random element in G_T. If $\mathfrak{b} = 0$, then let $T = e(g,g)^{abc}$. Otherwise, let $T = R_T$. Then, generate tuple (g, g^a, g^b, g^c, T), where $T = e(g,g)^{abc}$ or $T = R_T$. The *chal* will send (g, g^a, g^b, g^c, T) to $\mathcal{B}$, which acts as the challenger. A challenger access control policy Γ^* is sent to $\mathcal{B}$ [12].

Setup: The challenger *Chal* first runs algorithm $Setup(1^{\mu})$ to generate the key, keep the master key $MK = \{\alpha, \beta\}$, randomly choose $\beta' \in \mathbb{Z}_p^*$, and let $g^{\beta} = g^{b+\beta'} = g^b g^{\beta'}, g^a = g^{\alpha}$, where implicitly $\beta = b + \beta'$, $a = \alpha$. Then, $\mathcal{B}$ will obtain $e(g,g)^{\alpha\beta} = e(g,g)^{\alpha\beta'} e(g,g)^{ab}$. Finally, the $\{g, g^{\beta}, e(g,g)^{\alpha}, e(g,g)^{\alpha\beta}\}$ is sent to the adversary *Adv*.

Phase 1: Adv submits an attribute set $S \subseteq Att$ of the end-user to simulator $\mathcal{B}$, which does not meet the challenge access control policy Γ^*. Then, the algorithm $KeyGen(PK, MK, S)$ is run to generate the corresponding attribute key SK. As $g^{\beta} = g^{b}g^{\beta'}$ and $e(g,g)^{\beta} = e(g,g)^{\beta'}e(g,g^{b})$, the $\mathcal{B}$ queries the corresponding private key to the random oracle model, and get $D = g^{\alpha\beta}g^{(\alpha+\beta)r}g^{r^2e}$, $D' = e(g,g)^{-\beta r^2}$. For each $att_i \in S$, $\mathcal{B}$ selects random number r, $r_i' \in \mathbb{Z}_p^*$, implicitly set $r_i' = r_i + b$, $r' = r - b$. If $att_i \in \Gamma^*$, let $g^{r_i} = g^{b+r_i'}$. If $att_i \notin \Gamma^*$, let $g^{r_i} = g^{r_i'}$. Meanwhile, when $att_i \in \Gamma^*$, let $D_i = g^{r_i'+r'+H(att_i)}$, $D_i' = g^{b+r_i'}$. When $att_i \notin \Gamma^*$, let $D_i = g^{(r_i'+r'+H(att_i))}$, $D_i' = g^{r_i'}$. Finally, $\mathcal{B}$ sends the SK to the Adv.

Challenge: The Adv selects two context key plaintexts $ck_{i,0}, ck_{i,1}$ $(ck_{i,0} \neq ck_{i,1})$, which will be sent to the $\mathcal{B}$. The $\mathcal{B}$ randomly selects $\mathfrak{b} \in \{0,1\}$ and generates $CT^* = Encrypt(PK, ck_{i,\mathfrak{b}}, \mathcal{T}^*)$ based on $ck_{i,\mathfrak{b}}$. Then, the $\mathcal{B}$ returns CT^* to the Adv. Specifically, the simulator $\mathcal{B}$ sets $s_i = c$, $s_i' \in \mathbb{Z}_p^*$ and calculates the ciphertext with the help of the edge nodes. $\mathcal{B}$ obtains the complete ciphertext as follows in Equation (23).

$$\begin{aligned} CT^* = \{&\mathcal{T}^*, \widetilde{C} = ck_{i,\mathfrak{b}}e(g,g)^{\alpha\beta s_i}, \xi = g^{H(ck_{i,\mathfrak{b}})}, \\ &C = g^{s_i}, C' = g^{s_i - s_i'}, C'' = e(g,g)^{\alpha s_i} | i \in [1,\ell] \\ &\{\forall y \in Y, C_y = g^{q_y + r_y}, C_y' = g^{H(att(y)) + r_y}\}\} \end{aligned} \tag{23}$$

Then, the simulator $\mathcal{B}$ can construct the ciphertext form as follows in Equation (24).

$$\begin{aligned} \widetilde{C} &= ck_{i,\mathfrak{b}}e(g,g)^{\alpha\beta s_i} \\ &= ck_{i,\mathfrak{b}}e(g,g)^{a(\beta'+b)c} \\ &= ck_{i,\mathfrak{b}}e(g,g)^{a\beta' c}e(g,g)^{abc} \end{aligned} \tag{24}$$

Let $A = g^a$, $B = g^b$, and $g^{s_i} = g^c$. As $C'' = e(g,g)^{\alpha s_i}$ is known, the $e(g,g)^{a\beta' c}$ is computable.

Phase 2: Similar to Phase 1, the Adv will make inquiries in PPT. The limitation is that the inquiries in Phase 1 or the $ck_{i,0}$, $ck_{i,1}$ cannot be repeated.

Guess: The Adv outputs $\mathfrak{b}' = \{0,1\}$. If $\mathfrak{b}' = \mathfrak{b}$, the Adv outputs $T = e(g,g)^{abc}$, and the Adv wins the game. This advantage is $Pr[\mathcal{B}(g, g^a, g^b, g^c, T = e(g,g)^{abc})] = \frac{1}{2} + \epsilon$. Otherwise, the Adv outputs $T = R_T$, which is a random element in G_T. Namely, the challenge ciphertext CT^* is random in the view of the Adv. This advantage is $Pr[\mathcal{B}(g, g^a, g^b, g^c, T = R_T)] = \frac{1}{2}$. The probability of Adv winning this safety game $|Pr(\mathfrak{b}' = \mathfrak{b}) - \frac{1}{2}| = \frac{1}{2}(\frac{1}{2} + \epsilon + \frac{1}{2}) - \frac{1}{2} = \frac{\epsilon}{2}$. □

6.2. Performance Analysis

This section provides a performance analysis of DS-ABE-CC, systematically. The storage overhead of DS-ABE-CC was first compared with several schemes [6,8,17,23,27,34,35,37]. Then, the time overhead of some schemes was compared. Finally, we compared DS-ABE-CC with several advanced schemes for experiments. The performance analysis showed that DS-ABE-CC has good practicality and efficiency.

6.2.1. Time and Space Complexity Analysis

We define E_p as an exponential operation, and Ep_{G_i} represents the exponential operation in group G_1 (G_T). Let $Hash$ and $Pair$ represent the hash operation and bilinear mapping, respectively. A_u denotes the user attribute. ℓ is the access control level number. A_{c_i} denotes the attribute of the i-*th*-level-associated ciphertext CT. j represents the number of attributes included in the access policy in the corresponding schemes. N_T is the set of transfer nodes. S_i is the minimum internal node that satisfies the access structure. The hierarchical attribute related to the ciphertext is expressed as $A_c = \{A_{c_1}, \ldots, A_{c_\ell}\}$. $|M|$ denotes attribute the number in structure M.

Complexity analysis: We compared the complexity of the DS-ABE-CC scheme with some advanced schemes similar to this scheme, including Water's scheme [6], the BSW

scheme [8], Wang's scheme [17], Liu's scheme [34], Guan's scheme [27], the FAME Scheme [37], Tseng's scheme [23], and Yao's scheme [35]. These schemes are representative ones with low computational storage overhead, wherein Wang's scheme and Liu's scheme are hierarchical access control schemes, whose functions are more complex than other schemes, so their cost is relatively high. Table 2 shows the space and communication overhead comparison, whose data came from the schemes [6,8,17,23,27,34,35,37]. These schemes were built based on the asymmetric mapping ("MNT159", "MNT201", and "MNT224") and symmetric mapping ("SS512").

Table 2. Storage and communication overhead comparison.

Scheme	Access Structure	Key Size		Ciphertext Size	
		DO	EN	EN	DU
Our Scheme	Tree	$2\|\ell\|(\|G_1\|+\|G_T\|)$	$(2\|A_u\|+\ell)\|G_1\|+\|\ell\|(3\|G_1\|+2\|G_T\|)$	$3\|G_T\|$	$\|G_T\|+\|G_1\|$
Water's Scheme [6]	LSSS	$(\|A_u\|+2)\|G_1\|$	–	–	$(4\|A_c\|+1)\|G_1\|+\|G_T\|$
BSW Scheme [8]	Tree	$(2\|A_u\|+1)\|G_1\|$	–	–	$(2\|A_c\|+1)\|G_1\|+\|G_T\|$
Wang's Scheme [17]	Tree	$(2\|A_u\|+1)\|G_1\|$	–	–	$(2\|A_{c1}\|+\ell)\|G_1\|+(j\|N_T\|+k)\|G_T\|$
Liu's Scheme [34]	Tree	$(2\|A_u\|+1)\|G_1\|$	–	–	$(2\|A_{c1}\|+j\|N_T\|+\ell)\|G_1\|+(2j\|N_T\|+\ell)\|G_T\|$
Guan's Scheme [27]	And gate	$(\|A_u\|+\|Att\|)\|G_1\|+\|G_2\|$	–	–	$\|G_1\|+\|G_2\|+\|G_T\|$
FAME Scheme [37]	LSSS	$3(\|A_u\|+1)\|G_1\|+3\|H\|$	–	–	$(3\|A_c\|+1)\|G_1\|+\|G_T\|+3\|H\|$
Tseng Scheme [23]	ISSS	$(\|A_u\|+2)\|G_1\|$	–	–	$(4\|A_c\|+1)\|G_1\|+\|G_T\|$
Yao's Scheme [35]	LSSS	$(2\|A_u\|+2)\|G_1\|$	–	–	$(2\|A_c\|+2)\|G_1\|$

Ep is an exponential operation. Ep_{G_i} represents the exponential operation in group G_1 (G_T). *Hash* and *Pair* represent the hash operation and bilinear mapping, respectively. A_u denotes the user attribute. ℓ is the access control level number. A_{c_i} denotes the attribute of the i-*th*-level-associated ciphertext CT. $|M|$ denotes the attribute number in structure M. j represents the number of attributes included in the access policy. N_T is the set of transfer nodes. S_i is the minimum internal node that satisfies the access structure. The hierarchical attribute related to the ciphertext is expressed as $A_c = \{A_{c_1}, \ldots, A_{c_\ell}\}$. $|M|$ denotes the attribute number in structure M.

As shown in Table 3, we compared the DS-ABE-CC scheme with some schemes with low computational cost, among which Zhang's scheme [12] and Li's scheme [13] are advanced ABE schemes that support verifiable outsourcing. The comparison content of each scheme mainly includes the execution number of the exponential operation and pairing operation in the encryption phase *Encrypt* and the decryption phase *Decrypt*. For a more objective comparison, the time complexity can be analyzed by combining the time cost of each operation in Tables 4–7.

Table 3. Computation overhead comparison.

Scheme	Encrypt		Decrypt	
	DO	EN	EN	DU
Our Scheme	$2\ell Ep_{G_1}+2\ell Ep_{G_T}$	$(2\|A_c\|+\ell+1)Ep_{G_1}$	$\|A_u\|Ep_{G_T}+(4+\|A_u\|)Pair$	$2Ep_{G_T}$
Li's Scheme [13]	$4\,Ep_{G_1}+O(\|A_c\|\,Mutil\,)$	$9\|A_c\|Ep_{G_T}$	$(2+2\|A_u\|)Ep_{G_1}+(6+4\|A_u\|)Pair$	$Pair+4Ep_{G_T}$
Zhang's scheme [12]	$6Ep_{G_1}+O(\|A_c\|\,Mutil\,)$	$6\|A_c\|Ep_{G_T}$	$\|A_u\|Ep_{G_1}+(2+2\|A_u\|)Pair$	$Pair$
Water's Scheme [6]	$(3\|M\|+1)Ep_{G_1}+Ep_{G_T}$	–	–	$(2\|A_u\|+1)Pair+\|A_u\|Ep_{G_T}$
BSW Scheme [8]	$(2\|A_c\|+1)Ep_{G_1}+Ep_{G_T}$	–	–	$(2\|A_u\|+1)Pair+2\|S_i\|Ep_{G_T}$
Wang's Scheme [17]	$(2\|A_c\|+\ell)Ep_{G_1}+(j\|N_T\|+\ell)Ep_{G_T}$	–	–	$(2\|A_u\|+1)Pair+(\|S_i\|+j\|N_T\|+\ell)Ep_{G_T}$
Liu's Scheme [34]	$(2\|A_c\|+j\|N_T\|+\ell)Ep_{G_1}+(2j\|N_T\|+\ell)Ep_{G_T}$	–	–	$(2\|A_u\|+j\|N_T\|+\ell)Pair+(\|S_i\|+j\|N_T\|+\ell)Ep_{G_T}$
FAME Scheme [37]	$6\|A_c\|Ep_{G_1}$	–	–	$6Pair+(6\|A_u\|+3)Ep_{G_1}+6Ep_{G_T}$
Yao's Scheme [35]	$3\|A_c\|Ep_{G_1}$	–	–	$Pair+(6\|A_u\|+3)Ep_{G_1}+2Ep_{G_T}$

We neglected the lower cost *Mul* operation for the convenience of comparison. The definition of symbols here are the same as those in Table 2, as shown in the footnotes.

Table 4. Each operation in symmetric curve "SS512" (ms).

Group	Mul	Exp	Hash	Pairing
G_1	0.0041	0.0324	3.2	
G_2	0.0041	0.0324	3.2	0.838
G_T	0.001	0.038	—	

Table 5. Each operation in asymmetriccurve "MNT159" (ms).

Group	Mul	Exp	Hash	Pairing
G_1	0.0012	0.0121	0.0175	
G_2	0.0176	0.113	11.29	2.9
G_T	0.0048	0.305	—	

Table 6. Each operation in asymmetriccurve "MNT201" (ms).

Group	Mul	Exp	Hash	Pairing
G_1	0.0012	0.0147	0.155	
G_2	0.0239	0.213	21.751	5.62
G_T	0.0063	0.414	—	

Table 7. Each operation in asymmetriccurve "MNT224" (ms).

Group	Mul	Exp	Hash	Pairing
G_1	0.0016	0.0127	0.0479	
G_2	0.0169	0.1460	15.37	4.46
G_T	0.0047	0.292	—	

The relation of the time cost is as follows: $time(Pair) \gg time(Ep_{G_T}) \gg time(Ep_{G_1}) \gg time(Mutil)$. The DS-ABE-CC has a much lower time overhead and a trade-off space overhead compared to Water's scheme [6], the BSW scheme [8], Wang's scheme [17], Liu's scheme [34], Guan's scheme [27], the FAME scheme [37], Tseng's Scheme [23], and Yao's scheme [35].

6.2.2. Computational Performance Simulation

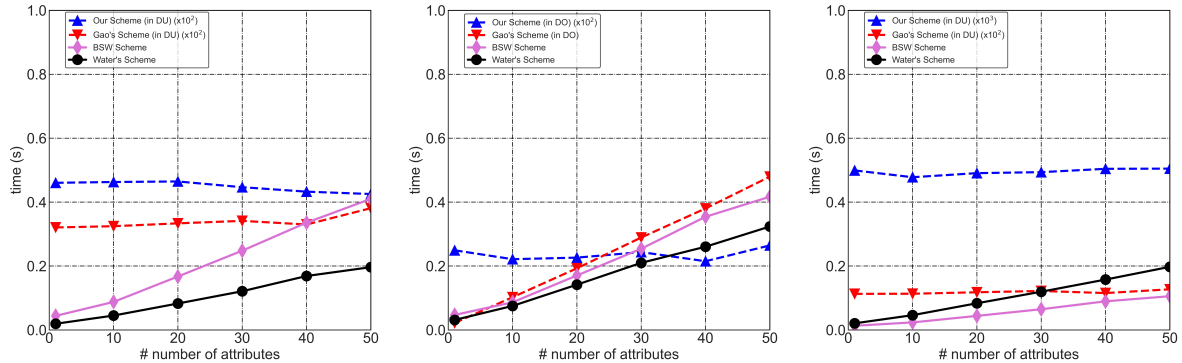
We conducted a comparative experiment to demonstrate the benefits of DS-ABE-CC. Our experiment was implemented on a 64-bit Ubuntu 16.4 virtual machine, using Python 3.7, Charm-0.43, and the PBC library associated with Charm-0.43. The host computer was a notebook computer with an Intel (R) Core (TM) i7-10510U CPU @ 1.80 GHz 2.30 GHz processor and 16 GB RAM. Because all Charm codes are designed under an asymmetric system, fortunately, relevant research results [11] have systematically proved that the principles, assumptions, and security proofs based on symmetric bilinear schemes can be converted into asymmetric settings in a general way, and they are completely equivalent:

(1) Time overhead for basic operation: We first conducted experiments on the time cost of the basic computational operations in bilinear mapping. Most existing schemes are designed based on symmetric and asymmetric bilinear mapping, so we chose four typical curves for implementation, including the symmetric curve "SS512" and the asymmetric curves "MNT159", "MNT201", and "MNT224". In the experiment, an element was randomly selected from the tested group G_1 (G_T), for exponentiation (Exp) and multiplication (Mul). The hashing $Hash$ denotes an operation that maps a value to a group element in G_1 (G_T). Each operation was performed 200 times, and the average value was taken. Multiplication (denoted by Mul) took the shortest time. Exponentiation (denoted by Exp) in G_2 had more cost than G_1 in general. The $Hash$ operation was only tested in group G_1 because all hash operations in the scheme were taken in group G_1. For comparison, we first executed an experiment on the symmetry curve, and the relevant experimental data in milliseconds are shown in Table 4. Then, we implemented it on the asymmetric curve, and the relevant data in milliseconds are shown in Tables 5–7. The time cost of various operations of asymmetric mapping ("MNT159", "MNT201", and "MNT224") was greater than that of symmetric mapping ("SS512"). "MNT159" is a relatively low-cost calculation in asymmetric mapping. It should be noted that the trend is consistent for different computer configurations, but there are differences in the specific experimental values.

(2) Time cost comparison in DOs and DUs: Zhang's scheme [12] and Li's scheme [13], like our DS-ABE-CC scheme, are Secure Data Sharing Scheme for Privacy-preserving in the IoT

and have excellent computing performance among similar solutions, with Zhang's scheme [12] having higher computing efficiency. To highlight the advantages of our DS-ABE-CC scheme, we compared the DS-ABE-CC scheme with several typical schemes, including the advanced and efficient outsourcing computing CP-ABE scheme (Zhang's scheme [12]), the CP-ABE scheme based on the access tree (BSW scheme [8]), and the CP-ABE scheme based on the LSSS (Water's scheme [6]). To obtain fair comparative experimental data, we set the simplest "and" gate access policy, with an access level $\ell = 1$ and an attribute number $|Att|$ (S) ranging from 1 to 50. We conducted several sets of classification experiments on symmetric curve $SS512$, to collect the time costs of the three key operations of *KeyGen*, *Encrypt*, and *Decrypt* in these schemes. The DO and DU in the BSW scheme [8] and Water's scheme [6] bear all the computational overhead of the three operations, while the computational overhead of the three operations in our DS-ABE-CC scheme and Zhang's scheme [12] is allocated to the DOs (DUs) and ENs.

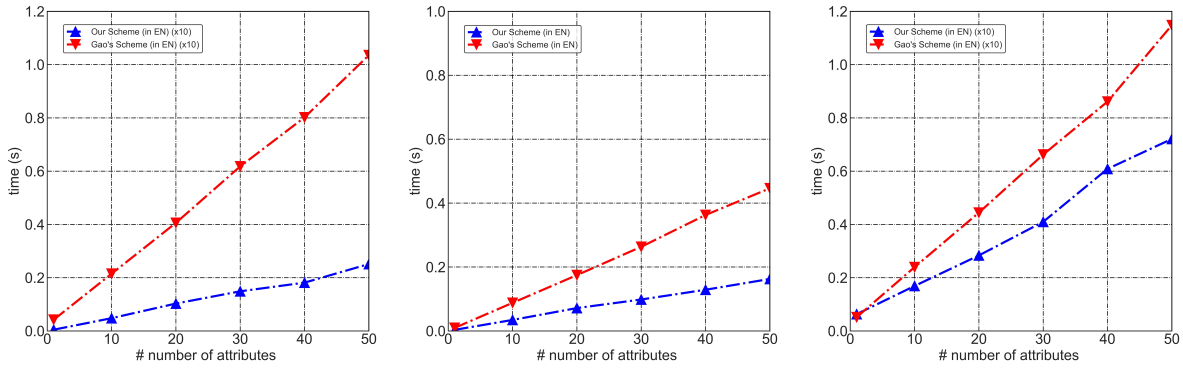
We increased $|S|$ from 1 to 50 in the symmetric curve "SS512", as shown in Figure 4. We expanded the experimental data of our scheme by 10^2 times or 10^3 times (marked with "$\times 10^2$" or "$\times 10^3$" in Figure 4) and placed the data in the same figure for comparison with the other schemes. In the *KeyGen* stage, the time cost for DUs (DOs) of our scheme and Zhang's scheme [12] basically remained constant, respectively about 0.004 s and 0.003 s. The time cost of our scheme was slightly greater than that of Zhang's scheme [12]. The time cost of the BSW scheme [8] and Water's scheme [6] increased about from 0.04 s to 0.4 s and about from 0.02 s to 0.2 s, respectively. Their time cost [6,8] was greater than our scheme. In the *Encrypt* stage, the time cost in the DOs of our scheme basically remained constant respectively, our scheme is 0.002s, while time cost of Zhang's scheme [12] increased from 0.02 s to 0.5 s. The time cost of the BSW scheme [8] and Water's scheme [6] increased about from 0.05 s to 0.5 s and from 0.03 s to 0.32 s, respectively. The time cost of our DS-ABE-CC scheme was approximately $\frac{1}{20}$ that of the BSW scheme [8] and Water's scheme [6]. As the attribute number increased, the time of our scheme remained unchanged, indicating that our scheme shifted some of the overhead of *Encrypt* to the edge nodes (ENs), improving the encryption efficiency of the terminal DO. In the *Decrypt* stage, the time cost in the DUs of our scheme and Zhang's scheme [12] basically remained constant, respectively approximately 0.0005 s and 0.001 s. The BSW scheme [8] and Water's scheme [6] increased from 0.01 s to 0.1 s and 0.02 s to 0.2 s, respectively. The time cost of our DS-ABE-CC scheme was approximately $\frac{1}{2}$ that of Zhang's scheme [12] and $\frac{1}{40} \sim \frac{1}{400}$ that of the BSW scheme [8] and Water's scheme [6]. As the number of attributes increased, the time overhead of the DS-ABE-CC scheme remained unchanged, indicating that DS-ABE-CC shifted some of the overhead of *Decrypt* to the edge nodes (ENs), improving the decryption efficiency of the terminal DUs. It can be predicted that when m continues to increase, the time cost will not increase in the same trend, which satisfies the lightweight demand for the IoT.



(**a**) *KeyGen* (**b**) *Encrypt* (**c**) *Decrypt*

Figure 4. Time cost comparison for DOs and DUs in "SS512".

(3)Time cost comparison in ENs: Lastly, we analyzed the computational overhead of our DS-ABE-CC scheme and Zhang's scheme [12] at the edge nodes. We increased $|S|$ from 1 to 50 in symmetric curve "SS512", as shown in Figure 5. We expanded the experimental data of our scheme by 10 times (marked with "$\times 10$" in Figure 5) and placed the data in the same figure for the comparison with the other schemes. In the *KeyGen* stage, the time cost on the ENs' side of our scheme and Zhang's scheme [12] increased linearly from 0.0004 s to 0.025 s and from 0.004 s to 0.11 s, respectively. The time cost on the ENs' side of Zhang's scheme [12] was greater than our scheme. As attributes increased, the more obvious the efficiency advantage of our DS-ABE-CC scheme was. In the *Encrypt* stage, the time cost on the ENs' side of our scheme and Zhang's scheme [12] increased linearly from 0.003 s to 0.16 s and from 0.01 s to 0.45 s, respectively. The time cost of our DS-ABE-CC scheme was approximately $\frac{1}{3}$ that of the Zhang's scheme [12]. As attributes increased, the time overhead at the edge node (EN) in our DS-ABE-CC scheme was also lower than in Zhang's scheme [12], as was the case on the DOs' side. In the *Decrypt* stage, the time cost on the ENs' side of our scheme and Zhang's scheme [12] increased linearly from 0.006 s to 0.07 s and from 0.005 s to 0.12 s, respectively. As attributes increased, the more obvious the efficiency advantage of our DS-ABE-CC scheme was.



(**a**) *KeyGen* (**b**) *Encrypt* (**c**) *Decrypt*

Figure 5. Time cost comparison for ENs in "SS512" .

To sum up, our DS-ABE-CC scheme had a lower time cost for the main steps than the three similar schemes, Zhang's scheme [12], the BSW scheme [8], and Water's scheme [6]. As the number of attributes increased, this advantage became more apparent. Strong evidence indicated that the DS-ABE-CC scheme we designed is based on a cloud–edge–terminal collaborative mechanism, which allocates the computing overhead of the terminal DUs' or DOs' side to the ENs without increasing the computing overhead of the edge nodes (ENs). The time cost of the DS-ABE-CC scheme on the DOs' or DUs' side was approximately only $\frac{1}{10} \sim \frac{1}{100}$ that of the other mainstream schemes, which greatly reduces the burden on the terminal and improves the user experience.

7. Conclusions

Based on the key distribution and agreement mechanism, this paper designed a data-sharing scheme of attribute-based encryption that supports cloud–edge–terminal collaborative computation (DS-ABE-CC). Firstly, on the premise of ensuring security, the DS-ABE-CC scheme outsources a large amount of overhead of the data owners and users (considered as "nodes") to untrusted edge nodes and the cloud, which significantly reduces the computing overhead of encryption and decryption, which is much less than the traditional scheme. Secondly, the DS-ABE-CC scheme considers the multiple owner/multiple user scenario of cloud–edge–terminal computation. It assigns a pair of private keys and public keys to each owner, user, and edge node, then embeds the key agreement in the encryption and decryption process, which greatly increases the security of the scheme. Finally,

the hierarchical access control for privacy preservation was designed without increasing the costs, and lightweight attribute and ciphertext updates were supported. The calculation cost of the *Encrypt* and *Decrypt* of DS-ABE-CC was lower than those of the traditional ABE schemes. In short, compared with other schemes, our DS-ABE-CC scheme made full use of the edge nodes to reduce the user computational costs and increase the security without increasing the overhead for edge nodes and deeply matched the IoT scenario where massive terminals access edge nodes. The advantages of the scheme included high security, high computing efficiency, and comprehensive functions.

Author Contributions: The conceptualization idea and methodology of the paper were proposed by K.Z. and C.D. conducted experimental simulations on the paper. J.W. reviewed the entire text. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Student Innovation Grant Program of the School of Cybersecurity.

Informed Consent Statement: Not applicable.

Data Availability Statement: The relevant data are available within the paper.

Acknowledgments: We express our gratitude to the relevant personnel from Beihang University who give us large help for our scheme.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following are some abbreviations for professional phrases that are used in this manuscript:

IoT	Internet of Things
CSP	Cloud service provider
DO	Data owner
DU	Data user
ABE	Attribute-based encryption
CP-ABE	Ciphertext-policy attribute-based encryption
CPA	Chosen-plaintext attack
DS-ABE-CC	A data sharing scheme of ABE supporting node–edge–cloud collaborative computation
LSSS	Linear secret-sharing scheme.

References

1. Song, D.X.D.; Wagner, D.; Perrig, A. Practical Techniques for Searches on Encrypted Data. In Proceedings of the 2000 IEEE Symposium on Security And Privacy, Berkeley, CA, USA, 14–17 May 2000; pp. 44–55.
2. Zheng, K.F.; Wang, N.; Liu, J.W. An efficient multikeyword fuzzy ciphertext retrieval scheme based on distributed transmission for internet of things. *Int. J. Intell. Syst.* **2022**, *37*, 7419–7443. [CrossRef]
3. Shamir, A. Identity-Based Cryptosystems And Signature Schemes. In *Proceedings of the Advances in Cryptology: Proceedings of CRYPTO 84*; Springer: Berlin/Heidelberg, Germany, 1985; pp. 47–53.
4. Boneh, D.; Franklin, M. Identity-Based Encryption from the Weil pairing. In Proceedings of the Advances in Cryptology—CRYPTO 2001: 21st Annual International Cryptology Conference, Santa Barbara, CA, USA, 19–23 August 2001; pp. 213–229.
5. Sahai, A.; Waters, B. Fuzzy identity-based encryption. In Proceedings of Advances in Cryptology–EUROCRYPT 2005: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, 22–26 May 2005; pp. 457–473.
6. Waters, B. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In Proceedings of the 14th International Conference on Practice and Theory In Public Key Cryptography Conference on Public Key Cryptography (PKC'11); Taormina, Italy, 6–9 March 2011; pp. 53–70.
7. Ning, J.T.; Huang, X.Y.; Wei, L.F. Tracing malicious insider in attribute-based cloud data sharing. *Chin. J. Comput.* **2022**, *45*, 1431–1445.
8. Bethencourt, J.; Sahai, A.; Waters, B. Ciphertext-policy attribute-based encryption. In Proceedings of 2007 IEEE Symposium on Security And Privacy (Sp'07), Berkeley, CA, USA, 20–23 May 2007; pp. 321–334.
9. Zhang, P.; Chen, Z.; Li, J. K. An efficient access control scheme with outsourcing capability and attribute update for fog computing. *Future Gener. Comput. Syst.* **2018**, *78*, 753–762. [CrossRef]
10. Wang, H.; He, D.; Shen, J. Verifiable outsourced ciphertext-policy attribute-based encryptionin cloud computing. *Soft Comput.* **2017**, *21*, 7325–7335. [CrossRef]

11. Xue, L.; Yu, Y.; Li, Y. Efficient Attribute-based Encryption with Attribute Revocation for Assured Data Deletion. *Inf. Sci.* **2019**, *479*, 640–650. [CrossRef]
12. Zhang, L.; Gao, X.; Mu, Y. Secure data sharing with lightweight computation in E-health. *IEEE Access* **2020**, *8*, 209630–209643. [CrossRef]
13. Li, Z.; Li, W.; Jin, Z. An efficient ABC scheme with verifiable outsourced encryption and decryption. *IEEE Access* **2019**, *7*, 29023–29037. [CrossRef]
14. Yan, X.X.; Meng, H. Ciphertext policy attribute-based encryption scheme supporting direct revocation. *J. Commun.* **2016**, *37*, 44–50.
15. Dong, G.F.; Lu, Y.K.; Zhang, C.W. CP-ABE key update method supporting revocation attribute. *Appl. Res. Comput.* **2022**, *40*, 142–149. [CrossRef]
16. Qiu, Z.; Zhang, Z.; Tan, S.; Wang, J.; Tao, X. Hierarchical Access Control with Scalable Data Sharing in Cloud Storage. *J. Internet Technol.* **2019**, *20*, 663–676.
17. Wang, S.; Zhou, J.; Yu, J.K. An Efficient File Hierarchy Attribute-Based Encryption Scheme in Cloud Computing. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 1265–1277. [CrossRef]
18. Shi, N.; Hou, Z.; Tan, M.; Shao, K.; Zhu, X. A threshold encryption scheme without a dealer based on Chinese remainder theorem. In Proceedings of the 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN), Guangzhou, China, 6–8 May 2017; pp. 90–96.
19. Liu, Z.; Cao, Z.; Huang, Q. Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles. In Proceedings of the Computer Security–ESORICS 2011: 16th European Symposium on Research in Computer Security, Leuven, Belgium, 12–14 September 2011; pp. 278–297.
20. Qian, H.; Li, J.; Zhang, Y. Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation. *Int. J. Inf. Secur.* **2015**, *14*, 487–497. [CrossRef]
21. Cui, H.; Wan, Z.; Deng, R.H. Efficient and expressive keyword search over encrypted data in cloud. *IEEE Trans. Dependable Secur. Comput.* **2016**, *15*, 409–422. [CrossRef]
22. Meng, R.; Zhou, Y.; Ning, J. An Efficient Key-Policy Attribute-Based Searchable Encryption In Prime-Order Groups. In Proceedings of the Provable Security: 11th International Conference, ProvSec 2017, Xi'an, China, 23–25 October 2017; pp. 39–56.
23. Tseng, Y.F.; Fan, C.I.; Liu, Z.C. Fast keyword search over encrypted data with short ciphertext in clouds. *J. Inf. Secur. Appl.* **2022**, *70*, 103320. [CrossRef]
24. Guo, F.; Mu, Y.; Susilo, W. CP-ABE with constant-size keys for lightweight devices. *IEEE Trans. Inf. Forensics Secur.* **2014**, *9*, 763–771.
25. Zhou, Z.; Huang, D. On efficient ciphertext-policy attribute based encryption and broadcast encryption. In Proceedings of the 17th ACM Conference on Computer and Communications Security, New York, NY, USA, 4 October 2010; pp. 753–755.
26. Doshi, N.; Jinwala, D.C. Fully secure ciphertext policy attribute-based encryption with constant length ciphertext and faster decryption. *Secur. Commun. Netw.* **2014**, *7*, 1988–2002. [CrossRef]
27. Guan, Z.; Yang, W.; Zhu, L.; Wu, L.; Wang, R. Achieving adaptively secure data access control with privacy protection for lightweight IoT devices. *Sci. China Inf. Sci.* **2021**, *64*, 162301. [CrossRef]
28. Green, M.; Hohenberger, S.; Waters, B. Outsourcing the Decryption of ABE Ciphertexts. In *Proceedings of the 2011 USENIX Conference on Security*; ACM: New York, NY, USA, 2016; pp. 1–16.
29. Mao, X.; Lai, J.; Mei, Q. Generic and Efficient Constructions of Attribute-Based Encryption with Verifiable Outsourced Decryption. *IEEE Trans. Dependable Secur. Comput.* **2016**, *13*, 533–546. [CrossRef]
30. Zhao, Z.; Wang, J. Verifiable outsourced ciphertext-policy attribute-based encryption for mobile cloud computing. *KSII Trans. Internet Inf. Syst. (TIIS)* **2017**, *11*, 3254–3272.
31. LI, J.; Huang, X.Y.; Li, J. Securely Outsourcing Attribute-Based Encryption with Checkability. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *28*, 2201–2210. [CrossRef]
32. Ostrovsky, R.; Sahai, A.; Waters, B. Attribute-based encryption with non-monotonic access structures. In Proceedings of the 14th ACM conference on Computer and communications security (CCS '07). Association for Computing Machinery, New York, NY, USA, 28 October 2007; pp. 195–203.
33. Pirretti, M.; Traynor, P.; McDaniel, P. Secure attribute-based systems. In Proceedings of the 13th ACM conference on Computer and communications security (CCS '06). Association for Computing Machinery, New York, NY, USA, 30 October 2006; pp. 99–112.
34. Liu, S.N.; Liu, B.; Guo, Z. File Hierarchy CP-ABE Scheme Supporting Graded User Access. *J. Softw.* **2022**, 1–14. [CrossRef]
35. Yao, X.; Chen, Z.; Tian, Y. A lightweight attribute-based encryption scheme for the Internet of Things. *Future Gener. Comput. Syst.* **2015**, *49*, 104–112. [CrossRef]
36. Miao, Y.; Ma, J.; Liu, X. Lightweight fine-grained search over encrypted data in fog computing. *IEEE Trans. Serv. Comput.* **2018**, *12*, 772–785. [CrossRef]
37. Agrawal, S.; Chase, M. FAME: Fast Attribute-based Message Encryption. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17), New York, NY, USA, 30 October 2017; pp. 665–682.