An Anonymous and Outsourcing-Supported Multiauthority Access Control Scheme With Revocation for Edge-Enabled IIoT System

Jie Cui, Fengyu Bian, Hong Zhong, Qingyang Zhang, Sheng Xu, Chengjie Gu, and Lu Liu

***Abstract*—With the application of the Internet of Things and intelligent technologies in industrial systems, the manufacturing efficiency and product quality have been improved, leading to the development of the Industrial Internet of Things (IIoT). Uploading private data to the cloud may lead to data leakage without security protection. Thus, attribute-based encryption (ABE) is widely used to ensure data security and implement data access control, and some multiauthority ABE schemes are proposed to meet the requirement of attributes from different authorities, such as factory and government, which is more suitable for the IIoT system. However, the current multiauthority ABE schemes are with privacy leakage problems on attributes and low-efficiency issues. To address these problems, we propose a multiauthority ABE scheme, which protects users' privacy by anonymizing attributes in authentication, reduces the computing burden of IIoT devices by adapting online/offline technique and outsourcing decryption to edge devices, and realizes effective attribute revocation. A formal security proof is presented that our scheme is replayable chosen ciphertext attack secure. Finally, we implement the proposed scheme, and experimental results show that our scheme is more efficient than the existing schemes.**

***Index Terms*—Attribute-based encryption (ABE), edge computing, Industrial Internet of Things (IIoT), multiauthority.**

I. Introduction

In recent years, the Industrial Internet of Things (IIoT) [1], [2] has attracted increasing attention in academia and industry. The IIoT devices are widely used to monitor, collect, and process data in the IIoT environment in real time, such as factory environment monitoring, smart home, intelligent vehicle, and healthcare. Through integrated sensing and monitoring functions of control sensors or controllers, mobile communications, intelligent analysis, and other technologies, the IIoT intelligently serves people. Due to the limited computing and storage capabilities of IIoT devices, collected data are usually sent to the cloud server and shared among authorized users. However, due to the vulnerability of IIoT devices and the sensitivity of data, it is necessary to consider the security and privacy issues involved in the process of uploading data. The simplest way is to directly encrypt data. However, the common encryption method is difficult for users to share data. At the same time, as the number of users accessing data increases, the process of accessing data becomes more complicated. Based on this problem, Sahai and Waters [3] introduced the concept of attribute-based encryption (ABE) to achieve fine-grained access control of encrypted data [4]–[6].

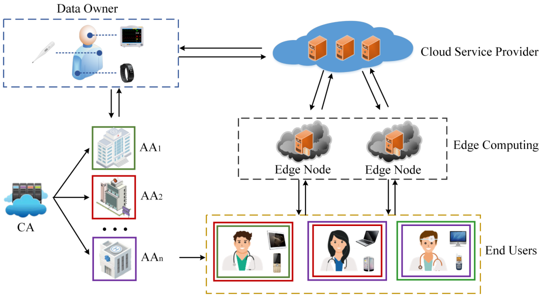

Fig. 1. System model of cloud-edge computing for the IIoT.

In the basic ABE scheme, an authority is required to manage a series of attributes and issue keys to the corresponding user; the ciphertexts can be succeeded in decrypting if user's attributes and access policies are matched successfully [7]. However, in many application scenarios, multiple authorities often manage attributes that may be from different domains [8]. Here, we take the healthcare IIoT (HealthIIoT) system as an example to illustrate. As shown in Fig. 1, the data owner (DO) wants to share the data with an end user (EU) who has the attribute of "doctor" distributed from a medical organization (AA_1), and the attribute "Medical Researcher" distributed by the manager of the clinical trial (AA_2). We can see another scenario that a patient from hospital AA_1 wants to be transferred to hospital AA_2, and the hospital AA_2 doctor can obtain the patient's health information as long as the attributes he holds successfully match the encryption access policy. Of course, there are similar scenarios in other

Manuscript received 25 July 2021; revised 8 November 2021, 19 January 2022, and 2 June 2022; accepted 3 July 2022. Date of publication 29 July 2022; date of current version 9 December 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62011530046, Grant 61872001, and Grant U1936220, in part by the Excellent Youth Foundation of Anhui Scientific Committee under Grant 2108085J31, and in part by the Special Fund for Key Program of Science and Technology of Anhui Province, China under Grant 202003A05020043. *(Corresponding author: Hong Zhong.)*

Jie Cui, Fengyu Bian, Hong Zhong, Qingyang Zhang, and Sheng Xu are with the Anhui Engineering Laboratory of IoT Security Technologies, School of Computer Science and Technology, and Institute of Physical Science and Information Technology, Anhui University, Hefei 230039, China (e-mail: cuijie@mail.ustc.edu.cn; 1772079142@qq.com; zhongh@ahu.edu.cn; qingyang.zhang.inchina@gmail.com; 1730384353@qq.com).

Chengjie Gu is with the Security Research Institute, New H3C Group, Hefei 230088, China (e-mail: gu.chengjie@h3c.com).

Lu Liu is with the School of Informatics, University of Leicester, LE1 7RH Leicester, U.K. (e-mail: l.liu@leicester.ac.uk).

Digital Object Identifier 10.1109/JSYST.2022.3189219

IIoT environments. For example, in a factory, the industrial chain may involve multiple factories. Although multiple authorities have brought many benefits to the IIoT, some challenges need to be solved in a multiauthority ABE (MA-ABE) scheme [9].

The challenging privacy issue needs to be addressed first. As shown above in a HealthIIoT environment, in order to protect the privacy of users, the hospital encrypts the medical information of each user with the user's name, ID, and department. However, the access policies are available to anyone who receives the ciphertext. In addition, when a patient from department AA_1 wants to be transferred to department AA_2, the attribute will be changed; this can be used to determine that the patient is highly likely to suffer from disease in department AA_2. In the above case, if the access policy is not protected, some sensitive information about patients may be disclosed. Therefore, the MA-ABE scheme must preserve the policy privacy. In addition, wearable devices, as a type of IIoT device, are used to collect patients' health data in real time. The collected health information needs to be encrypted and uploaded to the cloud. However, these processes involve many exponentiations and linear pair operations, which are linearly related to the number of attributes and the complexity of the access structure. Therefore, it is not appropriate for resource-constrained IIoT devices to perform these operations fast enough [10]. In order to solve this problem, the online/offline encryption technique [11] is introduced in the encryption process; it can be divided into two phases. In the offline phase, because the DO has not received the data and access structure, it first computes some complex operations based on the attributes of the patients, and after the data are generated, it encrypts the data based on the access structure with less computation overhead. Some complex encryption operations are performed in the offline phase, the online phase only requires a few computations, thus improving the efficiency in the encryption phase. At the same time, when the EU needs to use these data, it also faces the problem of computational efficiency. In order to ease the EUs' computing pressure, outsourced decryption [12] is proposed that most decryption computations are outsourced to proxy servers (PSs); hence, the resource-limited devices and EUs perform a limited number of computations. In recent years, with the emergence of edge computing [13], [14], many researchers have applied it to data outsourcing [15], [16], which relieves the computing pressure on users and improves efficiency, but they do not take into account the legitimacy of user identity in outsourced decryption. In addition, to meet the flexibility of the scheme, we must consider user/attribute revocation in the IIoT system. For example, the attributes of a left doctor should be revoked in time [17].

Combined with the above questions, it is a challenging task for edge-computing-enabled IIoT to design an MA-ABE that protects the privacy of access policy, reduces the computing overhead, and realizes revocation.

A. Related Work

In this part, we first introduce the current research progress of the IIoT based on MA-ABE and then analyze some problems that may be encountered in practical application.

1) ABE in the IIoT: Ciphertext-policy attribute-based encryption (CP-ABE) [18], [19] and MA-ABE [20] are applied to the IIoT environment.

The IIoT is a specific application of the Internet of Things (IoT), and the HealthIIoT system is often used in it. Yang *et al.* [21] proposed a scheme based on CP-ABE in the HealthIIoT environment, which effectively improved the quality of patient care. Yu *et al.* [19] designed an assured deletion scheme in the IIoT environment, which fulfills verifiable data deletion by utilizing CP-ABE. After, in order to realize data sharing in the IIoT environment, Miao *et al.* [18] presented a CP-ABE scheme.

To achieve better data sharing and management of attributes from different domains, more research works are turning to multiauthority schemes in the IIoT. Aiming to realize distributed access control of protected information in the IIoT, Yang *et al.* [20] designed a lightweight distributed data management scheme based on MA-ABE, which achieved data encryption, keyword secret door generation, and data recovery. To achieve flexible access control and solve the scalable key management problems in the HealthIIoT environment, Qian *et al.* [17] proposed an MA-ABE scheme. It could be regarded as an emerging health information exchange model, which helped the DOs to manage their data effectively. Later, in order to make the IIoT system more effective, Yan *et al.* [22] provided a rich expression ability MA-ABE scheme, which optimized the key structure, reduced the cost, and increased the flexibility. Recently, Fan *et al.* [23] have presented an outsourced MA-ABE scheme with privacy preserving, which can also be applied to the IIoT environment. The scheme introduced edge computing to implement low latency and outsourced transformations.

2) Efficiency and Privacy Issues: Following by the introduction on the MA-ABE in the IIoT, we review the privacy and efficiency ABE with online/offline encryption, outsourced decryption, and privacy protection in the IoT and the IIoT.

a) Online/offline cryptography: In order to reduce the computation overhead during encryption, Guo *et al.* [24] officially introduced an online/offline encryption technique, which divided the encryption process into offline and online phases. The offline phase did a lot of complex operations, while the online phase required only a few simple operations. Considering the computational overhead and the complexity of access policy, and the linear relationship between the number of attributes, Hohenberger and Waters [11] proposed an online/offline ABE scheme, which simply described the generation of the key in the online/offline phase. In 2016, an online/offline MA-ABE scheme was designed by Zhang *et al.* [25], but the key generation and computation cost of decryption were still high.

b) Outsourcing computation: Notice that some schemes' decryption is time consuming and inefficient. In 2011, the ABE-scheme-supported outsourcing was proposed by Green *et al.* [12]. Most decryption operations were outsourced to the proxy decryption server, and the user only needed to perform simple calculations. In 2013, Lai *et al.* [26] designed an ABE scheme, which supported verifiable outsourced decryption, but its encryption overhead was still high. In 2015, Qin *et* *al.* [27] designed an effective ABE scheme, which supported outsourced decryption to verify the correctness of the transformed ciphertext, but its encryption overhead was still high and the privacy protection was not considered. Later, Fan *et al.* [23] proposed an outsourced decryption MA-ABE scheme. The scheme increased the correctness of the outsourced decryption process, and reduced the time cost of decryption. However, the time cost of encryption was still high. Recently, Fan *et al.* [16] proposed a secure outsourced decryption MA-ABE scheme, but the privacy protection was not considered.

c) Policy preserving: In order to protect the access policy, in 2015, a CP-ABE scheme that adopted "AND" access structure was designed by Phuong *et al.* [28]. Unfortunately, only partial policy hiding could be supported. In 2017, Li *et al.* [29] proposed an MA-ABE scheme; it can realize both the privacy preserving and attribute revocation. However, it only protects the user privacy; the access policy privacy preserving cannot consider. Ma *et al.* [30] also proposed an MA-ABE scheme that realizes the policy privacy preserving and revocation. However, its decryption process involves multiple pairing operations, which are compute intensive and have high latency. In addition, the decryption time is also significantly increased with the increase in the number of attributes. In this case, it is difficult for resource-constrained IIoT devices to perform complex decryption operations quickly, so the scheme is unsuitable for IIoT. Later, according to the one-way anonymous protocol, an MA-ABE scheme that could protect the access policy was introduced by Zhong *et al.* [31]; although the scheme realized the attribute revocation, the computation overhead of encryption and decryption is very high. In 2019, Fan *et al.* [23] proposed an edge-computing-enabled MA-ABE scheme, which realized the hiding of access policy and user revocation, but the computation overhead was still very high in encryption, and it did not design an effective revocation algorithm to realize the update of key and ciphertext.

Although the above papers have done some work to protect the privacy, security, or efficiency of data in the IoT, they cannot be implemented efficiently at the same time, so it is a challenge for us to address the above issues in the IIoT system.

B. Our Contribution

To address aforementioned problems, we design an outsourced and revocable MA-ABE scheme for the IIoT, which is improved in terms of the security and efficiency compared with the existing schemes. The main contributions of this study are as follows.

1) By analyzing the security problem of the existing outsourced MA-ABE in the IIoT, we support online/offline and verifiable outsourced decryption in the proposed scheme, which could reduce user computing burden and ensure the legitimacy of user identity in the outsourced decryption.
2) To protect the privacy of user access policy, we support effective attribute transformation and revocation in the proposed scheme, which could increase the security and flexibility of the multiauthority CP-ABE scheme.
3) A formal security proof is presented that our scheme is replayable chosen ciphertext attack (RCCA) secure. In addition, we implement the proposed scheme, and experimental results show that our scheme is more efficient than the existing schemes.

C. Organization

The rest of this article is organized as follows. Section II introduces some related preparations. The system model, framework, and security model are introduced in Section III. The scheme construction is described in Section IV, followed by the correctness analysis and security proof in Section V. In Section VI, we evaluate the performance. Finally, Section VII concludes this article.

II. Preliminaries

A. Bilinear Maps

Definition 1: Let G_1 and G_2 be two multiplication cyclic groups with the same prime order p, while g is the generator of G_1. Thus, the bilinear map $e : G_1 \times G_1 \to G_2$ has three characteristics.

1) *Bilinearity:* $\forall w_1, w_2 \in G_1$, u and $v \in Z_p$, we have $e(w_1^u, w_2^v) = e(w_1, w_2)^{uv}$.
2) *Nondegeneracy:* $\exists\, g_1 \in G_1, e(g_1, g_1) \neq 1$.
3) *Computability:* $\forall w_1, w_2 \in G_1$, $e(w_1, w_2)$ can be computed efficiently.

B. Access Structure

Definition 2: Let $\{T_1, T_2 \ldots T_n\}$ be a series of parties, and collection Λ is a nonempty subsets of $\{T_1, T_2, \ldots, T_n\}$. Suppose that Λ is an access structure on $2^{\{T_1, T_2, \ldots, T_n\}}$. The collection Λ is monotone if it satisfies the property: $\forall B, C$: if $B \in \Lambda$, $B \subseteq C$, then $C \in \Lambda$. The sets in Λ are called the authorized sets. The sets not in Λ are unauthorized sets.

C. Linear Secret-Sharing Schemes

Definition 3: A linear secret-sharing scheme (LSSS) Π over a group of parties is defined as linear (over Z_p) if:

1) the shares for each party can structure a vector over Z_p;
2) there exists a share-generating matrix $M_{l \times n}$ for Π. $\forall i \in [1, l]$, the ith row of M is identified as the parties $\rho(i)$. For the column vector $\vec{v} = (s, r_2, \ldots, r_n)$, $s \in Z_p$ is the secret to be shared, and $r_2, \ldots, r_n \in Z_p$ are selected randomly; $M\vec{v}$ is the vector of l shares of the secret s according to Π; considering $\vec{\lambda} = (M \cdot \vec{v})_i$, λ_i belongs to party $\rho(i)$.

Suppose that there exists an LSSS Π for $M_{l \times n}$. Let $S \in M$ be any authorized set, and $I \subset [1, l]$ be defined as $I \subset \{i : \rho(i) \in S\}$. Then, there exist constants $\{\omega_i \in Z_p\}_{i \in I}$; if $\{\lambda_i\}$ are valid shares of any secret s according to Π, we can conclude that $\sum_{i \in I} \omega_i \lambda_i = s$.

III. System Model and Security Model

In this section, we present our system model and system framework in detail and also give the specific security model.

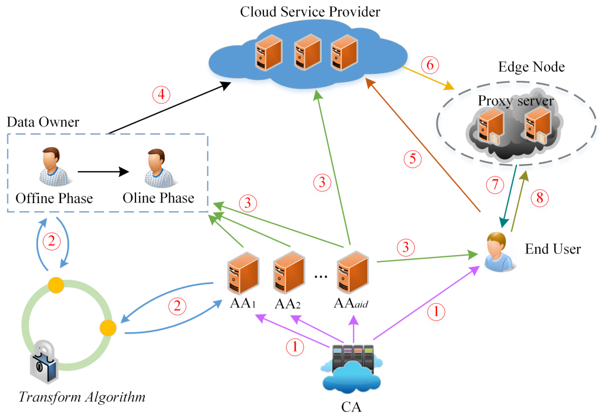

Fig. 2. System model of our scheme.

A. System Model

There are seven entities in our system model: global certification authority (CA), attribute authority (AA), DO, cloud service provider (CSP), edge node (EN), PS, and EU. As shown in Fig. 2, the CA mainly sets global common parameters and registers AA and EU. Then, it issues the identifier for each AA and the unique user identifier (step ①) for each EU.

Each AA has no intersection; it is primarily responsible for anonymizing attributes (step ②) and sending them to the relevant EU. It is also responsible for the generation and distribution of the keys within its own domain (step ③) and manages the update key generation for a nonrevoked user.

The DO first defines access policies over attributes and encrypts the data. Then, it uploads the encrypted data and anonymous attribute S' to CSP (step ④). Once a user's attribute is revoked, the DO needs to update the corresponding ciphertext component.

The CSP mainly stores the ciphertext from the DO and is responsible for managing anonymous attribute list and key list. It is also in charge of the request from EN (step ⑤) and updates the attribute list and the key list of revoked users. Here, the CSP is considered to be semitrusted, so both ciphertext and access policy need to be protected.

The EN is an entity between the CSP and the EU with limited storage and computing power. It mainly transfers and stores the data from the CSP, and the PS deployed on the EN is responsible for data outsourcing decryption and attribute authentication (step ⑥).

The EU obtains the key from the relevant AA (step ③). After the PS receives the data decryption request (step ⑥), it first obtains the key from the EU for partial decryption (step ⑧); then, the EU downloads the rest of the ciphertext from the PS (step ⑦) and recovers the encrypted data with its own secret key.

B. System Framework

Our revocable online/offline multiauthority ciphertext policy attribute-based encryption scheme with outsourced decryption (R-OO-MA-DO-CPABE) consists of the following algorithms.

$GlobalSetup(\lambda) \rightarrow (GP, aid, uid)$: This algorithm is performed by the CA. It first inputs security parameter λ; then, the global parameter GP, the unique identifier uid of the user, and the unique identifier aid of authority are outputted.

$AuthSetup(GP, aid) \rightarrow (SK_{aid}, PK_{aid}, ATK_{aid})$: The AA performs the authority setup algorithm. It first inputs GP, aid, and then outputs public key PK_{aid}, secret key SK_{aid}, and attribute transform key ATK_{aid}.

$AttrTrans(GP, ATK_{aid}, S) \rightarrow S'$: Both the AA and the DO are responsible for the implementation of attribute anonymity algorithm. The algorithm inputs GP, ATK_{aid}, and attribute set S; then, the transformed anonymous attribute S' is outputted.

$KeyGen(GP, uid, SK_{aid}, S, x) \rightarrow \{K_{x,uid}\}$: The AA performs the key generation algorithm, with the inputs GP, S, a uid, a secret key SK_{aid}, and attribute $x \in S$. It outputs private key $K_{x,uid}$ to the CSP for user identity authentication.

$KeyTransform(K_{x,uid}) \rightarrow \{ProxyK_{x,uid}, K_{uid}\}$: The key transform algorithm is performed by the EU. The algorithm inputs $K_{x,uid}$ and outputs the transformed proxy key $ProxyK_{x,uid}$ and the user recovery key K_{uid}.

$Offline.Enc(GP, PK) \rightarrow CT'$: The offline encryption algorithm is implemented by the DO, which outputs intermediate ciphertext by inputting GP, PK.

$Online.Enc(GP, CT', PK_{aid}, (M, \rho), m) \rightarrow CT$: The online encryption algorithm inputs data information, access matrix (M, ρ), middle ciphertext CT', public parameters GP, and the related public key PK_{aid} and outputs the ciphertext CT.

$UserAuthen(GP, CT, uid, K_{x,uid}) \rightarrow v_{\text{user}}$: The authentication algorithm is performed by the PS distributed on the EN. The algorithm inputs GP, ciphertext CT, a user uid, and private key $K_{x,uid}$ and outputs the algorithm authenticated result v_{user}.

$ProxyDec(GP, CT, v_{\text{user}}, ProxyK_{x,uid}) \rightarrow CT_{\text{out}}$: The proxy decryption algorithm is performed by the PS with the inputs GP, ciphertext CT, attribute authentication result v_{user}, and proxy key $ProxyK_{x,uid}$ and then outputs part decryption ciphertext CT_{out}.

$UserDec(GP, CT, CT_{\text{out}}, K_{uid}) \rightarrow m$: The EU runs the user decryption algorithm, which inputs GP, ciphertext CT, partially decrypts ciphertext CT_{out}, and user recovery key K_{uid} and outputs the final data information m.

$UserRev(L_K, L_{S_{\text{attr}}}, uid) \rightarrow (L'_K, L'_{S_{\text{attr}}})$: The user identity revocation algorithm is executed by the CSP. The algorithm inputs a user uid, key list L_K, and attribute list $L_{S_{\text{attr}}}$ and outputs the updated key and attribute list $(L'_K, L'_{S_{\text{attr}}})$.

$UKeyGen(S_R, SK_{aid}) \rightarrow UK_j$: The AA executed the key generation algorithm. The algorithm inputs revoked attribute sets S_R and the private key of the relevant authority SK_{aid}; then, the update key UK_j is outputted.

$SKUpdate(S_R, SK_{j,uid}, UK_j) \rightarrow \widetilde{K_{j,uid}}$: The key update algorithm is executed by the CSP. The algorithm inputs revoked attribute sets S_R, related private key $\underset{\sim}{K_{j,uid}}$, and the update key UK_j; then, the updated private key $\widetilde{K_{j,uid}}$ is outputted.

$CTUpdate(S_R, UK_j, CT) \rightarrow \widetilde{CT}$: The ciphertext update algorithm is implemented by the DO, which takes the revoked attribution set S_R, the update key UK_j, and the ciphertext CT as inputting and then outputs the updated ciphertext $\widetilde{CT}$.

C. Security Model

1) Security Model of the R-OO-MA-DO-CPABE Scheme: The concept of RCCA security was originally proposed by Canetti *et al.* [32], which can allow the modification of ciphertext but cannot change the underlying ciphertext. The notion of RCCA security is slightly stronger than chosen plaintext attack (CPA) security, and weaker than chosen ciphertext attack (CCA) security, but RCCA is enough for our scheme. Any ciphertext is not allowed to be modified in CCA, but can be supported in RCCA. Our scheme involves ciphertext transformation, so we adopt RCCA to prove security.

Now, let us think about security games R-OO-MA-DO-CPABE$_{\mathcal{A},\pi}(\lambda, U)$ for our scheme π, where U is the count of attributes. Here M, S, and S^* represent access structure, attribute set, and corrupted authority set, respectively. The game is interacting between adversary $\mathcal{A}$ and challenger $\mathcal{C}$. Let (I_{key}, I_{enc}) represent the input of $KeyGen, Encrypt$. Define a function $f : S \times M \to \{0, 1\}$, if $f(I_{key}, I_{enc}) = 1$ indicates that S satisfies M; otherwise, the result is 0.

Setup: The challenger $\mathcal{C}$ first runs the Setup algorithm, and then, it sends GP to the adversary $\mathcal{A}$.

Phase 1: The challenger $\mathcal{C}$ sets an empty table $\mathcal{T}$, an empty set $\mathcal{D}$, and the integer $\xi = 0$. We assume that $\mathcal{A}$ cannot make key queries if S^* satisfies M; then, it repeats the following query adaptively:

1) Create(I$_{key}$): First, the challenger $\mathcal{C}$ sets $\xi := \xi + 1$; then, it runs algorithm $KeyGen$ to get $SK = \{K_{x,uid}\}_{x \in I_{key}}$ and $PK = \{ProxyK_{x,uid}\}$ for some authenticated users and stores (ξ, I_{key}, SK, PK) in table $\mathcal{T}$. Finally, the proxy key PK is returned to $\mathcal{A}$.
2) Corrupt(i): If the ith entry exists in the table $\mathcal{T}$, then challenger $\mathcal{C}$ retrieves (i, I_{key}, SK) and sets the $\mathcal{D} = \mathcal{D} \cup \{I_{key}\}$; the private key SK is then returned to $\mathcal{A}$. If such entry does not exist in table $\mathcal{T}$, return $\perp$.
3) Decrypt(i, CT): If the ith entry exists in the table $\mathcal{T}$, the challenger $\mathcal{C}$ retrieves (i, I_{key}, SK, PK) and runs $(UserAuthen, ProxyDec, UserDec)$ algorithm. The result of the decryption algorithm is then returned; otherwise, return $\perp$.

Challenge phase: In this phase, the adversary $\mathcal{A}$ first selects randomly messages m_0 and m_1 with equal length; then, the adversary $\mathcal{A}$ gives a challenge access structure I^*_{enc} for all $I_{key} \in \mathcal{D}$ and $f(I_{key}, I^*_{enc}) \neq 1$. After that, the challenger $\mathcal{C}$ randomly selects a bit string of c and runs $Online.Enc(GP, Offline.Enc(GP), I^*_{enc}, m_c)$ to obtain ciphertext CT^*. Finally, the challenger $\mathcal{C}$ gives the ciphertext to $\mathcal{A}$. If $c = 0$, return (key^*, CT^*); if $c = 1$, and a session key R is selected from the key space, then return (R, CT^*).

Phase 2: Phase 1 can execute with the following conditions.

1) The adversary $\mathcal{A}$ cannot obtain the private key to challenge the ciphertext.
2) The adversary $\mathcal{A}$ cannot decrypt the message m_0 or m_1.

Guess: $\mathcal{A}$ outputs c' for c, only $c = c'$, and the game outputs 1.

Definition 4 (Security of revocable online/offline multiauthority CP-ABE with outsourced decryption): An online/offline multiauthority CP-ABE scheme with outsourced decryption is CCA secure. If the adversary $\mathcal{A}$ can win the game in the security model with a nonnegligible probability, then the following nonnegligible functional relation equation exists:

$$\Pr[\text{R-OO-MA-DO-CPABE}_{\mathcal{A},\pi}(\lambda, U) = 1] \leq \frac{1}{2} + \text{negl}(\lambda). \tag{1}$$

2) Security Model of Revocation: Here, we present the security model of revocation. We assume that the adversary can ask for two types of private keys of users: 1) the adversary could be a revoked user and the attributes of this user satisfy the policy of the challenge message; and 2) the adversary could be a nonrevoked user, and the attributes of this user do not satisfy the policy of the challenge message. Since this case has been taken into account by the above models, it will not be described in detail here.

Init: The adversary $\mathcal{A}$ first selects access structure (M^*, ρ^*) and revoked attribute set S_R.

Setup: The challenger $\mathcal{C}$ first runs the $AuthSetup$ algorithm to get PK, SK, if $x \in S_R$, it will update the PK, SK, and then, it sends PK to the adversary $\mathcal{A}$.

Phase 1: The adversary $\mathcal{A}$ can repeat the key query based on attribute set and user uid, where we assume that $\mathcal{A}$ cannot make key queries if $f(S^*, (M^*, \rho^*)) = 1$. Then, it selects some revocation users $\{uid_i\}^a_{i=1}$, where $f(S_R, (M^*, \rho^*)) = 1$. $\mathcal{C}$ runs algorithm $KeyGen, UKeyGen$ to get $K_{x,uid}$ and sends them to $\mathcal{A}$.

Challenge phase: The adversary $\mathcal{A}$ first randomly selects messages m_0 and m_1 with equal length, and it also submits the access structure (M^*, ρ^*) that $f(S^*, (M^*, \rho^*)) \neq 1$. After that, the challenger $\mathcal{C}$ randomly selects a bit string of $c \in \{0, 1\}$ and runs $Online.Enc$ on M_c to obtain ciphertext CT^*. Finally, the challenger $\mathcal{C}$ gives the ciphertext to $\mathcal{A}$.

Phase 2: Phase 1 can execute with the following condition.

1) The adversary $\mathcal{A}$ cannot get the key query for any attribute set if $f(S^*, (M_c, \rho^*)) = 1$.

Guess: The adversary $\mathcal{A}$ will output c' for c, only $c = c'$, and the game outputs 1.

Definition 5 (Security of revocable online/offline multiauthority CP-ABE with outsourced decryption): A revocable online/offline multiauthority CP-ABE scheme with outsourced decryption is to support revocation. If the adversary $\mathcal{A}$ can win the game in the security model with a nonnegligible probability, then the following nonnegligible functional relation equation exists:

$$\Pr[\text{R-OO-MA-CPABE}_{\mathcal{A},\pi}(\lambda, U) = 1] \leq \frac{1}{2} + \text{negl}(\lambda). \tag{2}$$

IV. System Construction

In this section, we will describe our specific scheme R-OO-MA-DO-CPABE; the detailed process is as follows.

A. System Setup

$GlobalSetup(\lambda) \to \{GP, aid, uid\}$: First, with the input of a security parameter λ, the algorithm selects two multiplication groups G_1 and G_2 with the prime order p and defines g as the

generator of G_1. Let $e : G_1 \times G_1 \rightarrow G_2$ be the bilinear map. Then, it selects a collision-resistant hash function $H : \{0,1\}^* \rightarrow G_1$, which maps global uid to each element in G_1. The CA then issues the corresponding identifier aid and identity identifier uid to each authority and each user, respectively. The global public parameter $GP = \{p, G_1, G_2, g, e, H\}$.

$AuthSetup(GP, aid) \rightarrow \{ATK_{aid}, SK_{aid}, PK_{aid}\}$: The S_{aid} is the attribute set managed by AA_{aid}, and any attribute set has no intersection; that is to say $S_{aid1} \cap S_{aid2} = \phi$, $aid1$ and $aid2$, respectively, represent two different institution identifier. For each attribute x in S_{aid}, AA_{aid} selects random numbers $\alpha_x, y_x \in Z_p$ to calculate $PK_{1,j} = e(g,g)^{\alpha_x}, PK_{2,j} = g^{y_x}$. Then, it publishes both public key $PK_{aid} = \{PK_{1,j}, PK_{2,j}\}$ and master secret key $SK_{aid} = \{\alpha_x, y_x\}$. Finally, it selects randomly two numbers $\varphi_{aid}, \psi \in Z_p$ and generates the attribute transformation key $ATK_{aid} = \{g^{\varphi_{aid}}, \psi\}$ for each AA; ATK_{aid} will be used in the attribute transformation algorithm.

B. Attribute Transformation

$AttrTrans(GP, ATK_{aid}, S) \rightarrow S'$: The attribute transformation algorithm takes the public parameter, the attribute transformation key, and an attribute set as input and then obtains the anonymous attribute S' for each attribute $x \in S$.

$$S' = \left\{x' | x' = e((g^{\varphi_{aid}})^{\psi}, H(x)) = e(g^{\varphi_{aid}}, H(x))^{\psi}\right\}. \quad (3)$$

C. Key Generation

$KeyGen(GP, uid, SK_{aid}, S, x) \rightarrow \{K_{x,uid}\}$: The authority AA will issue the corresponding private key for the EU who wants to access the data.

AA_{aid} first distributes the attribute set $S_{aid,uid} \subseteq S$ to the corresponding EU. Then, it runs the $AttrTrans$ algorithm to transform the attribute $x \in S_{aid,uid}$ as $S'_{aid,uid}$ and sends it to the CSP. $S'_{aid,uid}$ is added to the attribute list by the CSP, which can be described as $L_{S_{uid}} = L_{S_{uid}} \cup S'_{uid}$, and then distributed to the EU for storage. Finally, according to the corresponding user uid, AA_{aid} sets user private key as $K_{x,uid} = g^{\alpha_x} H(uid)^{y_x}$.

$KeyTransform(K_{x,uid}) \rightarrow \{ProxyK_{x,uid}, K_{uid}\}$: In this algorithm, the EU first selects a random number $z \in Z_p$; then, it computes the proxy key $ProxyK_{x,uid} = \{K_{x,uid}^{\frac{1}{z}}, H(uid)^{\frac{1}{z}}\}$ and sets the user recovery key as $K_{uid} = \{z\}$. It should be noted that the proxy key will be sent to the edge PS for partial decrypting, while the recovery key is held by the EU. At the same time, the proxy key $ProxyK_{x,uid}$ will also be sent to the CSP; the CSP adds key set to the key list L_K, which is set as $L_K = L_K \cup \{uid, ProxyK_{x,uid}, K_{x,uid}\}$.

D. Data Encryption

$Offline.Enc(GP, PK) \rightarrow \{CT'\}$: This stage is executed by the DO, and j is set from 1 to U, where U is the count of attributes. The algorithm first randomly selects $\lambda'_j, \alpha'_j, y'_j, \omega'_j, r_j \in Z_p$; then, for any j, there are the following calculations: $C_{1,j} = e(g,g)^{\lambda'_j} e(g,g)^{\alpha'_j r_j}, C_{2,j} = g^{r_j}, C_{3,j} = g^{y'_j r_j} g^{\omega'_j}$, $CT_{1,j} = PK_{1,j}^{r_j} \cdot e(g,g)^{-\alpha'_j r_j} = e(g,g)^{(\alpha_j - \alpha'_j) r_j}$,and $CT_{2,j} = PK_{2,j}^{r_j} \cdot g^{-y'_j r_j} = g^{(y_j - y'_j) r_j}$. Finally, the DO outputs the middle ciphertext

$$CT' = \{C_{1,j}, C_{2,j}, C_{3,j}, CT_{1,j}, CT_{2,j}\}_{j \in [1,U]}. \quad (4)$$

$Online.Enc(m, GP, PK_{aid}, (M, \rho), CT') \rightarrow \{CT\}$: In this stage, it is also performed by the DO. The algorithm inputs the message m, public parameter GP, the intermediate ciphertext CT', the public key PK_{aid}, and the matrix M. First, let S_T represent the set of attributes of the access policy T. Then, the DO runs the $AttrTrans$ algorithm to generate S'_T, and next, it generates an access matrix M of $l \times n$ with function ρ mapping its rows to the attributes. In addition, to encrypt m, the DO also selects a random number $s \in Z_p$, a vector $\vec{v} = (s, t_2, \ldots, t_n)^\top \in Z_p^n$, and a random vector $\vec{\omega} \in Z_p^n$ (0 is the first entry). Besides, let $\lambda_j = M_j \cdot \vec{v}$ and $\omega_j = M_j \cdot \vec{\omega}$; M_j represents the jth row of M and then calculates $C_0 = me(g,g)^s, C_{4,j} = \lambda_j - \lambda'_j, C_{5,j} = \omega_j - \omega'_j$, $D_{1,j} = e(g,g)^{\alpha_{\rho(j)} r_j}$, and $D_{2,j} = g^{y_{\rho(j)} r_j} g^{w_j}$. Finally, the DO returns the final ciphertext result as follows:

$$\begin{aligned} CT = \{&(M, \rho), C_0, C_{1,j}, C_{2,j}, C_{3,j}, C_{4,j} C_{5,j}, \\ &CT_{1,j}, CT_{2,j}, D_{1,j}, D_{2,j}\}_{j \in [1,U]}. \end{aligned} \quad (5)$$

E. Data Decryption

$UserAuthen(GP, CT, uid, K_{x,uid}) \rightarrow v_{uid}$: After the CSP receives the data access request from an EU with uid, it checks the relevant anonymous attribute set S_{uid} from $L_{S_{uid}}$ and then sends S_{uid} to the PS. To prevent illegal users from using the attribute set, the PS selects a constant $c_j \in Z_p^*$, satisfying $\sum_x c_j M_j = (1, 0, \ldots, 0)$ to authenticate S_{uid} and then calculate

$$v_{\text{user}} = \prod_j \left(\frac{D_{1,j} e(H(uid), D_{2,j})}{e(K_{\rho(j),uid}, C_{2,j})} \right)^{c_j}. \quad (6)$$

$v_{\text{user}} = 1$ means that for each $x \in S_{uid}$ matches the legitimate user uid, then the PS will perform the decryption algorithm. If $v_{\text{user}} \neq 1$, the PS outputs $\perp$, and the decryption algorithm will stop immediately.

$ProxyDec(CT, ProxyK_{x,uid}, GP, v_{\text{user}}) \rightarrow CT_{\text{out}}$: The PS executes the decryption algorithm with $v_{\text{user}} = 1$. For $j \in 1, 2, \ldots l$, the decryption algorithm calculates $C'_{1,j} = C_{1,j} \cdot CT_{1,j} \cdot e(g,g)^{C_{4,j}}, C'_{2,j} = C_{2,j}, C'_{3,j} = C_{3,j} \cdot CT_{2,j} \cdot g^{C_{5,j}}$; then, according to $c_j \in Z_p^*, \sum_x c_j M_j = (1, 0, \ldots, 0)$, it computes

$$CT_1 = \prod_j^l \left(\frac{e\big(H(uid)^{\frac{1}{z}}, C'_{3,j}\big)}{e\big(K_{j,uid}^{\frac{1}{z}}, C'_{2,j}\big)} \right)^{c_j} \quad (7)$$

$$CT_2 = \prod_j^l (C'_{1,j})^{c_j}. \quad (8)$$

In the end, the partial decrypted ciphertext $CT_{\text{out}} = \{CT_1, CT_2\}$ are sent to the EU.

$UserDec(GP, CT, CT_{\text{out}}, K_{uid}) \rightarrow m$: Once the EN receives CT_{out}, it first calculates $CT'_2 = CT_2^{\frac{1}{z}}, CT = CT'_2 \cdot CT_1$ and then calculates $CT^z = e(g,g)^s$. Finally, the EN recovers the message with user key K_{uid}:

$$m = C_0 / CT^z. \quad (9)$$

Algorithm 1: $SKUpdate$.

```
Input: $S_R, K_{j,uid}, UK_j$
if $a_j \notin S_R$ then
    $\widetilde{K_{j,uid}} = g^{\alpha_j} H(uid)^{y_j}$
    $\widetilde{ProxyK_{j,uid}} = \{K_{j,uid}^{\frac{1}{z}}, H(uid)^{\frac{1}{z}}\}$
else
    $\widetilde{K_{j,uid}} = K_{j,uid} \cdot g^{UK_j}$
    $\widetilde{ProxK_{j,uid}} = \{\widetilde{K_{j,uid}^{\frac{1}{z}}}, H(uid)^{\frac{1}{z}}\}$
end if
return $\widetilde{K_{j,uid}}, \widetilde{ProxK_{j,uid}}$
```

Algorithm 2: CTU_{pdate}.

```
Input: $S_R, CT, UK_j$
if $a_j \notin S_R$ then
    $\widetilde{CT_{1,j}} = CT_{1,j}$
    $\widetilde{D_{1,j}} = D_{1,j}$
else
    $\widetilde{CT_{1,j}} = CT_{1,j} \cdot e(C_{2,j}, g^{UK_j})$
    $\widetilde{D_{1,j}} = D_{1,j} \cdot e(C_{2,j}, g^{UK_{\rho(j)}})$
end if
return $\widetilde{CT_{1,j}}, \widetilde{D_{1,j}}$
```

F. User Revocation

When a user's identity needs to be revoked, there is no need to update the key or re-encrypt the ciphertext. Instead, the revoked uid only needs to be sent to the CSP, and it will delete the revoked user key. Without the relevant key, the decryption and recovery of the original data cannot be carried out. The specific process of user identity revocation is as follows.

$UserRev\{L_K, L_{S_{\text{attr}}}, uid\} \rightarrow \{L'_K, L'_{S_{\text{attr}}}\}$: According to the corresponding uid, the CSP will delete the key $\{ProxyK_{aid,uid}, K_{x,uid}\}$ and the attribute set S_{uid} from L_K and $L_{S_{\text{attr}}}$. Then, it stores updated L'_K and $L'_{S_{\text{attr}}}$.

G. Attribute Revocation

Here, let S_R represent attribute set of user uid', who needs to revoke attribute from authority AA_{aid}. In order to prevent the revoked user from being decrypted again, all the unrevoked users with attribute set S_R can change their stored data with the following three algorithms.

1) $UKeyGen(S_R, SK_{aid}) \rightarrow UK_j$: This algorithm is run by AA_{aid}. AA_{aid} first selects a random $\alpha_j^* \in Z_p$; then, it computes the proxy update key $UK_j = \alpha_j^* - \alpha_j$. Then, it updates public key $\widetilde{PK_{1,j}} = e(g,g)^{\alpha_j^*}$ and secret key $\widetilde{SK_{1,j}} = \alpha_j^*$ for each attribute belongs to S_R, while the other (PK_{aid}, SK_{aid}) remains unchanged. Finally, AA_{aid} sends UK_j to the unrevoked user and the DO under a secure channel.

2) $SKUpdate(S_R, K_{j,uid}, UK_j) \rightarrow \{\widetilde{K_{j,uid}}, \widetilde{ProxyK_{j,uid}}\}$: When the CSP receives UK_j from AA_{aid}, the $SKUpdate$ algorithm will be implemented to update the private key and the proxy key for revocation attribute a_j.

3) $CTUpdate(S_R, CT, UK_j) \rightarrow \widetilde{CT}$: After the DO receives the update key UK_j from AA_{aid}, for $a_j \in S_R$, ciphertext $CT_{1,j}$, and $D_{1,j}$ need to be updated, while other ciphertexts remain unchanged.

V. Theoretical Analysis

In this section, we will analyze our scheme, including the correctness analysis and security proof.

A. Correctness Analysis

1) Data Decryption: We first calculate the value of intermediate decryption ciphertext and then verify the correctness of the equation by obtaining the corresponding value as follows:

$$\begin{aligned} C'_{1,j} &= C_{1,j} \cdot CT_{1,j} \cdot e(g,g)^{C_{4,j}} \\ &= e(g,g)^{\lambda'_j} e(g,g)^{\alpha'_j r_j} e(g,g)^{(\alpha_j - \alpha'_j) r_j} e(g,g)^{\lambda_j - \lambda'_j} \\ &= e(g,g)^{\lambda_j} e(g,g)^{\alpha_j r_j} \end{aligned} \tag{10}$$

$$C'_{2,j} = C_{2,j} = g^{r_j} \tag{11}$$

$$\begin{aligned} C'_{3,j} &= C_{3,j} \cdot CT_{2,j} \cdot g^{C_{5,j}} \\ &= g^{y'_j r_j} g^{\omega'_j} \cdot g^{(y_j - y'_j) r_j} \cdot g^{\omega_j - \omega'_j}. \end{aligned} \tag{12}$$

If the attribution is owned by the same user, and the identity identifier is matched, then compute

$$\begin{aligned} CT_1 = &\prod_j^l \left(\frac{e(H(uid)^{\frac{1}{z}}, C'_{3,j})}{e(K_{j,uid}^{\frac{1}{z}}, C'_{2,j})} \right)^{c_j} = \prod_j^l \left(\frac{e(H(uid), g)^{\frac{\omega_j}{z}}}{e(g,g)^{\frac{r_j \alpha_j}{z}}} \right)^{c_j} \\ &= \prod_j^l \left(\frac{e(H(uid), g)^{\frac{\omega_j}{z}}}{e(g,g)^{\frac{r_j \alpha_j}{z}}} \right)^{c_j} = \frac{1}{e(g,g)^{\sum \frac{c_j r_j \alpha_j}{z}}} \end{aligned} \tag{13}$$

$$\begin{aligned} CT_2 &= \prod_j^l (C'_{1,j})^{c_j} = \prod_j^l (e(g,g)^{\lambda_j} e(g,g)^{\alpha_j r_j})^{c_j} \\ &= e(g,g)^s e(g,g)^{\sum \alpha_j r_j c_j}. \end{aligned} \tag{14}$$

Therefore, the following equations can be obtained:

$$CT'_2 = CT_2^{\frac{1}{z}} = e(g,g)^{\frac{s}{z}} e(g,g)^{\sum \frac{c_j r_j \alpha_j}{z}} \tag{15}$$

$$\begin{aligned} CT &= CT'_2 \cdot CT_1 = e(g,g)^{\frac{s}{z}} C_0 / CT^z \\ &= m e(g,g)^s / (e(g,g)^{\frac{s}{z}})^z \\ &= m. \end{aligned} \tag{16}$$

2) Attribute Authentication: In the previous, we have made ω_j represent $M_j \cdot \vec{v}$, so we can see that the $\omega \cdot (1, 0, \ldots, 0) = 0$; in the case of attribution $x \in S_{uid}$ matching the user uid, we can get the following authentication:

$$\begin{aligned} v_{\text{user}} &= \prod_j \left(\frac{D_{1,j} e(H(uid), D_{2,j})}{e(K_{\rho(j),uid}, C_{2,j})} \right)^{c_j} \\ &= \prod_j \left(\frac{e(g,g)^{\alpha_{\rho(j)} r_j} e(H(uid), g^{y_{\rho(j)} r_j} g^{\omega_j})}{e(g^{\alpha_{\rho(j)}} H(uid)^{y_{\rho(j)}}, g^{r_j})} \right)^{c_j} \\ &= \prod_j \left(\frac{e(g,g)^{\alpha_{\rho(j)} r_j} e(H(uid), g^{y_{\rho(j)} r_j} g^{\omega_j})}{e(g,g)^{\alpha_{\rho(j)} r_j} e(H(uid), g)^{y_{\rho(j)} r_j}} \right)^{c_j} \\ &= \prod_j e(H(uid), g)^{\omega_j c_j} = 1. \end{aligned} \tag{17}$$

B. Security Proof

In this part, we indicate that our study is RCCA secure, and we also demonstrate that our scheme satisfies the privacy protection of access policy.

To prove our scheme satisfying RCCA security, we consider the security of the revocable online/offline multiauthority CP-ABE. We can directly base the security of our online/offline system on that of the underlying Lewko–Waters (LW) [33] system. Based on the static assumption in composite order groups, the LW system can be reduced to selectively CPA security. In [12], it has described how to construct a RCCA-secure scheme from a selectively CPA one. Our scheme can be proved to be RCCA secure in a similar way.

1) CPA Security of the Revocable Online/Offline Multiauthority CP-ABE (R-OO-MA-CPABE) Scheme: Before proving, we also define the R-OO-MA-CPABE scheme security game as R-OO-MA-CPABE$_{\mathcal{A},\pi}(\lambda, U)$, and the specific process is similar to the process of game R-OO-MA-DO-CPABE. Similarly, Definition 6 can be obtained as follows.

Definition 6 (Security of R-OO-MA-CPABE): An online/offline multiauthority attribute encryption scheme meets the requirements of CCA security. If $\mathcal{A}$ has a nonnegligible advantage to win the game in the security model, then the following nonnegligible functional relation equation exists:

$$\begin{aligned} &\Pr[\text{R-OO-MA-CPABE}_{\mathcal{A},\pi}(\lambda, U) = 1] \\ &\qquad \leq \frac{1}{2} + \text{negl}(\lambda). \end{aligned} \tag{18}$$

Theorem 1: The R-OO-MA-CPABE mentioned above is selectively CPA secure with reference to Definition 6 assuming that LW [33] is selectively CPA secure.

Proof: We will demonstrate that for any polynomial probability time (PPT) adversary $\mathcal{A}$ with a non-negligible advantage can break the selective of CPA security of LW scheme with a PPT simulation algorithm $\mathcal{B}$. The $\mathcal{B}$ behaves like a challenger interacting with $\mathcal{A}$ in game R-OO-MA-CPABEA$_{\mathcal{A},\pi}(\lambda, U)$.

Init: First of all, adversary $\mathcal{A}$ gives algorithm $\mathcal{B}$ the S and $(M^*_{l\times n}, \rho^*)$; then, $\mathcal{B}$ passes S, (M^*, ρ^*) to LW challenger.

Setup: Then, $\mathcal{C}$ provides GP to $\mathcal{B}$ and sends it to $\mathcal{A}$.

Phase 1: When $\mathcal{A}$ publishes any key generation query, in order to get the private key $K_{x,uid}$, algorithm $\mathcal{B}$ sends request to the LW challenger $\mathcal{C}$; then, $\mathcal{B}$ returns $K_{x,uid}$ to $\mathcal{A}$.

Challenge: The $\mathcal{B}$ sends two different messages m_0 and m_1 that randomly selected in LW message space to the LW challenger. The LW challenger randomly selects a bit string $c \in \{0, 1\}$ and encrypts m_c under (M_c, ρ^*). Then, it sends back the challenge ciphertext of message m_c, such that $CT = ((M_c, \rho^*), C_0, \{C_{1,j}, C_{2,j}, C_{3,j,}, CT_{1,j}, CT_{2,j}, C_{4,j}, C_{5,j}, D_{1,j}, D_{2,j}\})$. The LW challenger then selects some random values, $\hat{\lambda}_j, \hat{\omega}_j, \hat{\alpha}_j, \hat{y}_j \in Z_p$ and calculates the ciphertext

$$\begin{aligned} C^*_{1,j} &= C_{1,j} e(g,g)^{-\hat{\lambda}_j} e(C_{2,j}, g^{-\hat{\alpha}_j}) \\ &= e(g,g)^{(\lambda_j - \hat{\lambda}_j)} e(g,g)^{(\alpha^j - \hat{\alpha}_j) r_j} \end{aligned} \tag{19}$$

$$C^*_{2,j} = C_{2,j} \tag{20}$$

$$\begin{aligned} C^*_{3,j} &= C_{3,j} / (C_{2,j}^{\hat{y}_j} g^{\hat{\omega}_j}) \\ &= g^{(y_j - \hat{y}_j) r_j} g^{\omega - \hat{\omega}_j} \end{aligned} \tag{21}$$

$$CT^*_{1,j} = e(C_{2,j}, g^{\hat{\alpha}_j}) = e(g,g)^{\hat{\alpha}_j r_j} \tag{22}$$

$$CT^*_{2,j} = C_{2,j}^{\hat{y}_j}, C^*_{4,j} = \hat{\lambda}_j, C^*_{5,j} = \hat{\omega}_j \tag{23}$$

$$D^*_{1,j} = D_{1,j}, D^*_{2,j} = D_{2,j}. \tag{24}$$

All random values in decryption phase can cancel out each other; the algorithm $\mathcal{B}$ guesses which message $m_{c'}$ $(c' \in 0, 1)$ is encrypted and outputs the result $key_{guesss} := C/m_{c'}$. Finally, it sends the challenge ciphertext $CT^* = \{(M^*, \rho), C_0, C^*_{1,j}, C^*_{2,j}, C^*_{3,j}, CT^*_{1,j}, CT^*_{2,j}, C^*_{4,j}, C^*_{5,j}, D^*_{1,j}, D^*_{2,j}\}$ and key_{guesss} to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ makes key queries adaptively and $\mathcal{B}$ proceeds as Phase 1.

Guess: In the end, $\mathcal{A}$ outputs a bit c'. If $c' = 0$, $\mathcal{B}$ outputs c'; if $c' = 1$, $\mathcal{B}$ outputs $1 - c'$. This distribution is perfect for $\mathcal{A}$, and if $\mathcal{A}$ wins the R-OO-MA-CPABE game we proposed with an distinguish advantage, then the LW scheme can also be broken by $\mathcal{B}$ with the same advantage. □

2) RCCA Security of the R-OO-MA-DO-CPABE Scheme: Now, we prove that our scheme satisfies the RCCA security. We have showed that RCCA security is better than CPA, because our decryption phase involves the ciphertext transformation, so we use notion of CCA security.

In this part, we first give the structure of the RCCA; it can be analogous to scheme [12], while $Setup$, $KeyGen$, $Offline.Enc$, and $KeyTransform$ phases are as before. In addition to the public parameters part, we define a hash function $H_1 : \{0, 1\}^* \to \{0, 1\}^k$ and then describe our algorithm.

Online.Enc($GP, m, (M, \rho)$*)*. In the beginning of this phase, the random number R is selected by encryption algorithm from G_T and calculates $s = H(R, m), r = H_1(R)$. The values of $(\{C_{1,j}, C_{2,j}, CT_{1,j}, CT_{2,j}, C_{3,j}, C_{4,j}, C_{5,j}, D_{1,j}, D_{2,j}\})$ have been calculated as before; it then computes $C = R \cdot e(g,g)^s, C'' = m \oplus r$ and sends them in the ciphertext.

UserAuthen(·): This process authenticates the user's legitimacy.

ProxyDec(·): The algorithm calculates the value of CT_1 and CT_2 and sends the ciphertext value $CT_{\text{out}} = (C, C'', CT_1, CT_2)$ to PS.

UserDec(·): In this process, the decryption algorithm first takes $ProxyK_{x,uid}, K_{x,uid}$, and ciphertext CT_{out} as input and then computes $CT = CT_1 \cdot CT_2^{\frac{1}{z}}, R = C/CT^z, m = C'' \oplus H_1(R), s = H(R,m)$. If $C = R \cdot e(g,g)^s, CT = e(g,g)^{s/z}$, the algorithm outputs m, otherwise $\perp$.

Theorem 2: If the proposed R-OO-MA-CPABE scheme is selectively CPA secure, then according to Definition 4, our scheme satisfies RCCA security.

Proof: To prove Theorem 2, we assume that there is an adversary that can break our proposal with ϵ probability. Then, we construct a simulation algorithm $\mathcal{B}$ to attack R-OO-MA-CPABE with the probability of ϵ minus the negligible amount.

Init: The algorithm $\mathcal{B}$ runs $\mathcal{A}$, and $\mathcal{A}$ returns the access structure (M^*, ρ^*) to $\mathcal{B}$; $\mathcal{B}$ then sends it to the R-OO-MA-CPABE challenger as the structure it wants to be challenged.

Setup: For all attribute x, $\mathcal{B}$ gets the public parameters $GP = \{g, PK_{1,j}, g^{y_j}\}$ from the R-OO-MA-CPABE scheme and sends it to $\mathcal{A}$.

Phase 1: $\mathcal{B}$ sets an empty set $\mathcal{D}$, empty table $\mathcal{T}, \mathcal{T}_1, \mathcal{T}_2$ and $\xi = 0$. Then, one of the adversary's queries will be answered.

1) $H(R,m)$: If the tuple (R,m,s) is in $\mathcal{T}_1$, return s. Else, randomly select s from Z_p and add the (R,m,s) to the table $\mathcal{T}_1$, then return s.
2) $H_1(R)$: If the tuple (R,r) is in $\mathcal{T}_2$, return r. Else, randomly select $r \in \{0,1\}^k$ and add the (R,r) to the table $\mathcal{T}_2$, then return r.
3) Creat(S): The algorithm $\mathcal{B}$ sets $\xi = \xi + 1$, then one of the following will be done.

a) If S satisfies (M^*, ρ^*), it selects a transformation key by selecting a random e from Z_p and then runs $KeyGen(uid, x, SK_{aid}, GP)$, $x \in S$, to obtain SK'. It sets $SK = (e, PK)$, $PK = SK'$ and selects e to fit the distribution of PK.

b) If not the above case, the key generation oracle model will be called to obtain SK and then randomly select sets $PK = ProxyK_{x,uid}, x \in S$ and $z \in Z_p$. Finally, it stores (x, S, SK, PK) in the table $\mathcal{T}$ and sends PK to $\mathcal{A}$.

i) Corrupt(x): The algorithm $\mathcal{B}$ retrieves whether entry (x, S, SK, PK) exists; if it exists, $\mathcal{B}$ sets $\mathcal{D} = \mathcal{D} \cup S$ and returns $\mathcal{A}$; otherwise, SK returns $\perp$.

ii) Decrypt(x, CT): Both $\mathcal{A}$ and $\mathcal{B}$ can perform the transformation operation, because they all have access to PK. Let ciphertext $CT = (C, C'', CT_1, CT_2)$, which is related to (M^*, ρ^*), and if (x, S, SK, PK) exists, we can get from the table $\mathcal{T}$. If not or $S \notin (M^*, \rho^*)$,
we return $\perp$ to $\mathcal{A}$.

If x does not satisfy (M^*, ρ^*):

1) $Parse\, SK = (z, PK), R = C/CT_1^z \cdot CT_2$ is computed;
2) get the tuple (R, m_x, s_x) from T_1; if not, return $\perp$;
3) In this set, if indices $i \neq j$, there does not exist (R, m_i, s_i) and (R, m_j, s_j) in $\mathcal{T}_1$, and $m_i = m_j,\ s_i = s_j$, then algorithm $\mathcal{B}$ aborts;
4) get (R, r) from T_2; else the algorithm $\mathcal{B}$ aborts;
5) check whether the equation $C = Re(g,g)^{s_x}, C'' = m_x \oplus r,\ CT_1 CT_2^{\frac{1}{z}} = e(g,g)^{\frac{s_x}{z}}$ is established for x;
6) if x can pass all of the above tests, then output the message m_x, otherwise output $\perp$;

If x satisfies (M^*, ρ^*):

1) parse $SK = (e, PK)$, $\beta = CT_2 CT_1^e$ is computed;
2) if (R, m_x, s_x) exists in table $\mathcal{T}_1$, check whether the equation $\beta = e(g,g)^{s_x}$ is established;
3) if it does not match, $\mathcal{B}$ outputs $\perp$;
4) if more than one match is successful, the algorithm terminates;
5) otherwise, only (R, m_x, s_x) can match successfully;
6) get (R, r) from table $\mathcal{T}_2$; if none exists, $\mathcal{B}$ terminates.
7) calculation equation $C = Re(g,g)^s$, $CT_1^e CT_2 = e(g,g)^s$, $C'' = m \oplus r$.
8) output m when all the checks are correct, otherwise $\perp$.

Challenge. Finally, $\mathcal{A}$ outputs $(m_0, m_1) \in \{0,1\}^{2k}$, $\mathcal{B}$ does the following.

1) $\mathcal{B}$ randomly selects message $(M_0, M_1) \in G_T^2$, passes it to the challenger of our scheme, and obtains a ciphertext $CT = (C, \{C_{1,j}, C_{2,j}, C_{3,j}\})$ with (M^*, ρ^*).
2) $\mathcal{B}$ randomly selects $C'' \in \{0,1\}^k$.
3) It will send $CT^* = (C, C'', \{C_{1,j}, C_{2,j}, C_{3,j}\})$ to the adversary $\mathcal{A}$.

Phase 2: $\mathcal{B}$ answers the query as in Phase 1; if the decryption query is M_0^* or M_1^*, it outputs the test message.

Guess: $\mathcal{A}$ has to either terminate the algorithm or output a string; either way, $\mathcal{B}$ will be ignored. $\mathcal{B}$ searches the tables $\mathcal{T}_1$ and $\mathcal{T}_2$ to check if M_0, M_1 is the first element to appear in any entry. If neither value appears, $\mathcal{B}$ outputs a bit string as a guess. If only M_c appears, $\mathcal{B}$ outputs c as a guess.

If a correct guess is outputted by adversary $\mathcal{A}$, it implies that it knows M_c with ϵ probability; in other words, $\mathcal{A}$ can query M_c with ϵ probability, and the ciphertext CT transmitted by R-OO-MA-CPABE can be decrypted by $\mathcal{B}$ with $\epsilon + \text{negl}(\lambda)$ probability. The result shows that if the $\mathcal{A}$'s advantage of R-OO-MA-CPABE$_{\mathcal{A},\pi}(\lambda, U)$ is ϵ, then the advantage of $\mathcal{A}$ in our scheme is $\epsilon - \text{negl}(\lambda)$.

Theorem 3: Assuming that the construction scheme of LW meets the requirements of CPA security, then, according to Definition 4, our proposed proposal also meets the requirements of CPA security.

Proof: RCCA security includes CPA security, so Theorem 3 is valid according to the above two theorems. □

3) Security of the R-OO-MA-CPABE Scheme: To prove the revocation of the scheme, we first prove the following theorem.

Theorem 4: Based on the static assumption in composite order groups, the construction scheme of LW meets the requirements of CPA security; then, our proposed proposal also meets the revocation security.

Proof: To prove Theorem 4, we assume that there is an adversary that can break our proposal with ϵ probability, Then, we construct a simulation algorithm $\mathcal{B}$ to attack the LW scheme with the probability of ϵ minus the negligible amount, and $\mathcal{C}$ is the challenger that interacts with $\mathcal{B}$ in the LW scheme.

Setup: The adversary $\mathcal{A}$ first sends access structure (M^*, ρ^*) and S_R to $\mathcal{B}$, and then, it sends them to $\mathcal{C}$. For $x \in S - S_R, \mathcal{C}$ sets $\widetilde{PK}_{1,j} = e(g,g)^{\alpha_j^*}$. For $x \in S_R$, $\mathcal{C}$ sets $\widetilde{PK}_{1,j} = e(g,g)^{\alpha_j}$ ·

$UK_j = e(g,g)^{\alpha_j^*}$. Then, it sends $PK_{1,j} = \widetilde{PK_{1,j}}$ to adversary $\mathcal{A}$.

Phase 1: The adversary $\mathcal{A}$ can repeat the key query based on attribute set and user uid; then, $\mathcal{B}$ sends the attribute set and user uid to $\mathcal{C}$. The $\mathcal{C}$ runs algorithm $KeyGen, UKeyGen$ to get $K_{x,uid}, UK_j$ and sends them to the $\mathcal{A}$.

Challenge: In this phase, the adversary $\mathcal{A}$ first selects randomly messages m_0 and m_1 with equal length, and it also submits the access structure (M^*, ρ^*) that $f(S^*, (M^*, \rho^*)) \neq 1$. After that, the challenger $\mathcal{C}$ randomly selects a bit string of $c \in \{0, 1\}$ and runs $Online.Enc$ on M_c to obtain ciphertext CT^*. Finally, the challenger $\mathcal{C}$ gives the ciphertext to $\mathcal{B}$. For $x \in S_R$, $\widetilde{CT_{1,j}} = CT_{1,j} \cdot e(C_{2,j}, g^{UK_j})$, $\widetilde{D_{1,j}} = D_{1,j} \cdot e(C_{2,j}, g^{UK_{\rho(j)}})$. Then, it sends back the challenge ciphertext of message m_c, such that $CT = ((M_c, \rho^*), C_0, \{C_{1,j}, C_{2,j}, C_{3,j,}, \widetilde{CT_{1,j}} = CT_{1,j}, CT_{2,j}, C_{4,j}, C_{5,j}, \widetilde{D_{1,j}} = D_{1,j}, D_{2,j}\})$.

Guess: In the end, $\mathcal{A}$ outputs a bit c'. If $c' = 0$, $\mathcal{B}$ outputs c'; if $c' = 1$, $\mathcal{B}$ outputs $1 - c'$. This distribution is perfect for $\mathcal{A}$, and if $\mathcal{A}$ wins the R-OO-MA-CPABE game we proposed with an distinguish advantage, then the LW scheme can also be broken by $\mathcal{B}$ with the same advantage. □

Security of access policy: When the DO wants to send the encrypted data to the cloud, some sensitive information can be easily exposed, because the access policy is explicitly sent with ciphertext. To protect the privacy of the policy, it needs to protect the attribute x of access policy. Therefore, we adopt a transformation algorithm based on a one-way anonymous key agreement protocol [34] to implement the anonymization. It makes attribute x anonymous to $x' = e((g^{\varphi_{aid}})^{\psi}, H(x))$ before the DO encrypts the data and the AA sends the attributes to the EU. Because of the randomness of φ, only both AA and authorized users with $H(x)^{\varphi_{aid}}$ can calculate x', while the CPS and the EN cannot guess x from x'. Because users cannot get any attribute information, they cannot get any access policy information even if they collude with each other. □

VI. Performance Evaluation

In this section, we will evaluate the performance of the proposed scheme and several existing ABE schemes from the functions and the computation overhead perspective. Considering the characteristics that our scheme satisfies, we choose the schemes [8], [23], [28], [31], [35] for comparison. We mainly analyze whether these schemes support LSSS access structure, multiauthority, online/offline encryption, outsourced decryption, and attribute revocation functions. Then, we compare the encryption, decryption, and ciphertext update time of the scheme [8], [23], [31], [35] through experiments.

A. Functional Comparison

In Table I, we compare with the previous ABE scheme in terms of access structure (AS), the number of authority (A), online/offline encryption (OE), outsourced decryption (OD), privacy protection (PP), and attribute revocation (AR). We notice that our proposal is multiple authorities, supporting LSSS access structure, outsourced decryption, privacy preserving, and attribute revocation. Therefore, other schemes are not as flexible as ours.

TABLE I
Comparison of Flexibility

Schemes	AS	A	OE	OD	PP	AR
Phuong's [28]	And	Single	No	No	Yes	No
R's [8]	LSSS	Multiple	No	No	No	No
Shao's [35]	LSSS	Multiple	Yes	Yes	Yes	No
Fan's [23]	LSSS	Multiple	No	Yes	Yes	No
Zhong's [31]	LSSS	Multiple	No	No	Yes	Yes
Our scheme	LSSS	Multiple	Yes	Yes	Yes	Yes

TABLE II
Notations Used for Performance Comparison

Notations	Meaning
E_1/E_T	An exponentiation in the group G_1/G_T
M_1/M_T	An multiplication in the group G_1/G_T
M_{Z_p}	An multiplication in Z_p
S_{Z_p}	An sum in Z_p
c	The number of attributes related to ciphertext
e	The pairing operation
$\|G_1\|/\|G_T\|/\|Z_p\|$	The length in the group $G_1/G_T/Z_p$
$\|S_k\|/\|S_c\|$	The attribute set length in key generation and encryption

B. Experiment Analysis

Generally speaking, it is possible to convert the assumption and security proof to asymmetric settings in a general manner, and higher levels of security can be achieved, so we convert our scheme to asymmetric settings [29], [36]. We have selected three groups of G_1, G_2, and G_T and an asymmetric bilinear map $e : G_1 \times G_2 \to G_T$ in the experiment. We implement all the schemes using BN254 elliptic curve relied on $y^2 = x^3 + 5$, and it is based on the Miracl core library and Mosaic library. The experiments are evaluated in the Ubuntu 18.04.4LTS system with Intel Core CPU@3.40 GHz and 8-GB RAM. In Tables II and III, we give the experimental performance comparison notations and analyze the encryption and decryption time overhead of five schemes. In Table IV, we compare the size of the public parameters, ciphertext, and key of the five schemes. Generally, the experimental encryption time is mainly composed of two operations: exponential operation and multiplication operation. In our experiment, the multiplication calculation time of G_1 and G_T was 0.004 and 0.032 ms, respectively. The exponential operation time of G_1 and G_T was 0.816 and 3.483 ms, respectively, and the time of linear pairing operation was 8.063 ms. (Because the calculation on G_2 in the experiment is included in G_T, the calculation on G_2 is not considered here.) In order to avoid the contingency of the results, we repeat the experiment 1000 times under each access policy and then calculate the final result with the average value.

Figs. 3–5 compare the time cost of the schemes [8], [23], [31], [35]; the number of attributes of each authority is set from 1 to 20. Figs. 3 and 4 describe the encryption and decryption times that vary with the number of attributes in each authority. In addition, in Fig. 6, we also compare the ciphertext update time with revocable attributes' number changing, which is set from 1 to 20. During encryption and decryption, we generate ciphertext

TABLE III
COMPUTING OVERHEAD COMPARISON OF FIVE SCHEMES

Schemes	Operations(Encry)	Operations(Decry)
R's [8]	$E_T + M_T + (3E_1 + 2M_{Z_p} + M_1 + 2E_T + M_T)c$	$(3E_T + 3M_T + E_1)c + M_T$
Fan's [23]	$2E_T + E_1 + M_T + (5E_1 + 2M_1 + 4M_{Z_p} + E_T)c$	$E_T + M_T$
Zhong's [31]	$E_T + E_1 + M_T + (2E_T + 3E_1 + M_T + M_1 + 3M_{Z_p})c$	$(2e + 2M_T + E_T)c + M_T$
Shao's [35]	Online: $M_T + E_T + E_1 + (4S_{Z_p} + 2M_{Z_p})c$ Offline: $(2E_T + 3E_1 + M_{Z_p} + M_T + M_1)c$	$E_T + 2M_T$
Our scheme	Online: $E_T + M_T + 2E_1 + (2S_{Z_p} + M_T + M_1 + 2M_{Z_p} + 2E_1)c$ Offline: $(4E_T + 4E_1 + 2M_T + M_1 + S_{Z_p} + 4M_{Z_p})c$	$2E_T + 2M_T$

TABLE IV
PARAMETER SIZE COMPARISON

Scheme	PP	SK	Ciphertext
R's [8]	$\lvert G_1\rvert$	$2\lvert S_k\rvert\lvert G_1\rvert$	$\lvert G_T\rvert + (\lvert G_T\rvert + 3\lvert G_1\rvert)\lvert S_c\rvert$
Fan's [23]	$\lvert G_1\rvert + \lvert G_T\rvert$	$\lvert Z_p\rvert + 4\lvert S_k\rvert\lvert G_1\rvert$	$\lvert G_T\rvert + \lvert G_1\rvert + (\lvert G_T\rvert + 3\lvert G_1\rvert)\lvert S_c\rvert$
Zhong's [31]	$\lvert G_1\rvert + \lvert G_T\rvert$	$2\lvert S_k\rvert\lvert G_1\rvert$	$\lvert G_T\rvert + \lvert G_1\rvert + (\lvert G_T\rvert + 2\lvert G_1\rvert)\lvert S_c\rvert$
Shao's [35]	$\lvert G_1\rvert + \lvert G_T\rvert$	$2\lvert S_k\rvert\lvert G_1\rvert$	offline:$(\lvert G_T\rvert + 2\lvert G_1\rvert)\lvert S_c\rvert$ online:$(\lvert G_T\rvert + \lvert G_1\rvert + 4\lvert Z_p\rvert)\lvert S_c\rvert$
Our scheme	$\lvert G_1\rvert + \lvert G_T\rvert$	$2\lvert S_k\rvert\lvert G_1\rvert$	offline:$(2\lvert G_T\rvert + 3\lvert G_1\rvert)\lvert S_c\rvert$ online:$(2\lvert G_T\rvert + \lvert G_1\rvert + 2\lvert Z_p\rvert)\lvert S_c\rvert$

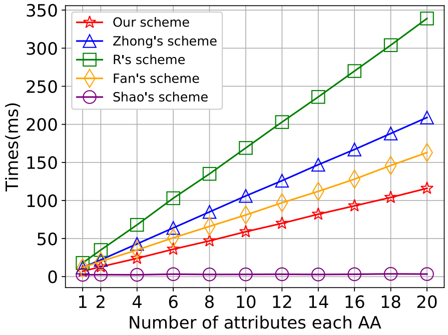

Fig. 3. Comparison of encryption time.

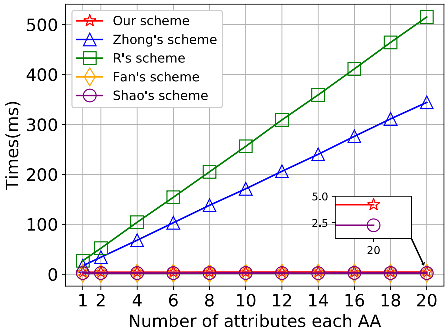

Fig. 4. Comparison of decryption time.

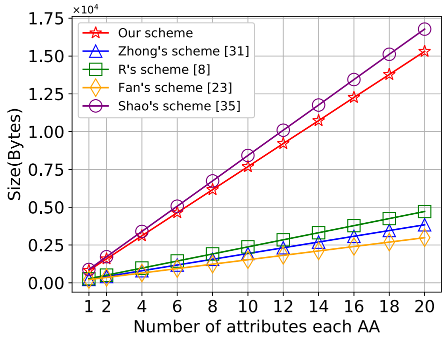

Fig. 5. Comparison of the size of ABE ciphertext.

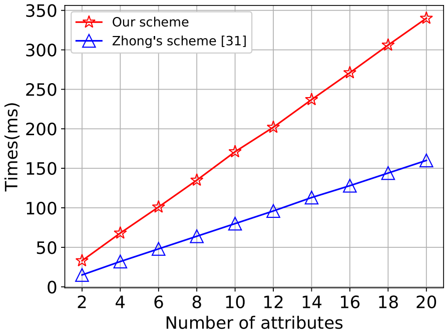

Fig. 6. Comparison of ciphertext update time.

under the "AND" access structure to maintain the consistency of the attributes; the number of attributes is set from 1 to 20.

1) Encryption Time: Fig. 3 shows that there is a linear growth relationship between the encryption time and the attributes. Compared with the schemes in [8], [23], [31], and [35], we can see that when the number of attributes increases to 20, the encryption times of the schemes in [8], [23], [31], and [35] are 340.072, 163.629, 209.287, and 3.3 ms, respectively, while our encryption time is 115.153 ms. It is pretty obvious that our scheme and the scheme in [35] have significantly lower encryption time cost than the schemes in [8], [23], and [31], mainly because we both adopt the online and offline encryption technologies, which put complex computing operations into the offline stage. Because we generate two authenticated ciphertexts in the encryption phase, the encryption overhead is slightly higher. However, our scheme implements revocation, while the scheme in [35] does not.

2) Decryption Time: Similarly, from Fig. 4, we can see that when the number of attributes increases to 20, the encryption times of the above schemes are 515.147, 2.108, 344.757, and 2.571 ms, respectively, while our decryption time is 3.5 ms. It can be noticed that our decryption overhead is significantly lower than that of the schemes in [8] and [31], while the schemes [23] and [35] are not significantly different from our overhead, mainly because they all use outsourced decryption technology, which alleviates the computing pressure of EUs.

3) ABE Ciphertext Size: Fig. 5 shows the change of the ABE ciphertext communication overhead with the number of attributes in [8], [23], [31], and [35]. It can be seen that the communication cost of our scheme is lower than that of the scheme in [35], but higher than that of schemes in [8], [23], and [31]. This is mainly because we introduce intermediate ciphertext in the encryption process to complete the user identity authentication function in the outsourced decryption process, which is also a reasonable cost considering the functions performed by the scheme.

4) Update Ciphertext Time: Fig. 6 shows that the ciphertext update overhead of our proposal is slightly higher than that of the scheme in [31]. This is because in the process of ciphertext updating, our scheme needs to update the ciphertext $D_{1,j}(j \in [1, U])$, which is used for outsourcing decryption. The scheme in [31] does not support outsourcing decryption technology, so the overhead time of our scheme is slightly doubled. However, the overhead time of our scheme in encryption and decryption is largely lower than that of the scheme in [31]. In addition, since the attribute revocation does not require frequent operations, it is a reasonable time overhead given the functionality provided by our solution.

In a word, there is less computational overhead of our scheme in encryption and decryption phases by comparison and effectively realizes the ciphertext update.

VII. Conclusion

In this article, we presented a secure multiauthority CP-ABE scheme to achieve fine-grained access control of data in the IIoT environment. Our scheme adopted an anonymous algorithm based on a one-way anonymity protocol to realize privacy protection. To relieve the burden on users and ensure the security of the data, most encryption operations were carried out in the offline phase, delegating complex authentication and decryption operations to ENs. In addition, we proposed a revocation method for our study that can be applied to IIoT environments in practice. Moreover, the security analysis and experiments showed that our scheme is effective and flexible.

References

[1] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.

[2] J. Cui, F. Wang, Q. Zhang, Y. Xu, and H. Zhong, "An anonymous message authentication scheme for semi-trusted edge-enabled IIoT," *IEEE Trans. Ind. Electron.*, vol. 68, no. 12, pp. 1330–1334, Dec. 2021.

[3] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2005, pp. 1330–1334.

[4] H. Xiong, H. Zhang, and J. Sun, "Attribute-based privacy-preserving data sharing for dynamic groups in cloud computing," *IEEE Syst. J.*, vol. 13, no. 3, pp. 2739–2750, Sep. 2019.

[5] K. Xue, N. Gai, J. Hong, D. Wei, P. Hong, and N. Yu, "Efficient and secure attribute-based access control with identical sub-policies frequently used in cloud storage," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 1, pp. 1330–1334, Jan./Feb. 2022.

[6] J. Hong et al., "TAFC: Time and attribute factors combined access control for time-sensitive data in public cloud," *IEEE Trans. Serv. Comput.*, vol. 13, no. 1, pp. 158–171, Jan./Feb. 2020.

[7] J. Wei, W. Liu, and X. Hu, "Secure and efficient attribute-based access control for multiauthority cloud storage," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1731–1742, Jun. 2018.

[8] Y. Rouselakis and B. Waters, "Efficient statically-secure large-universe multi-authority attribute-based encryption," in *Proc. Int. Conf. Financial Cryptography Data Secur.*, 2015, pp. 1330–1334.

[9] K. Xue et al., "RAAC: Robust and auditable access control with multiple attribute authorities for public cloud storage," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 4, pp. 953–967, Apr. 2017.

[10] S. J. De and S. Ruj, "Efficient decentralized attribute based access control for mobile clouds," *IEEE Trans. Cloud Comput.*, vol. 8, no. 1, pp. 124–137, Jan.–Mar. 2020.

[11] S. Hohenberger and B. Waters, "Online/offline attribute-based encryption," in *Proc. Int. Workshop Public Key Cryptography*, 2014, pp. 1330–1334.

[12] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in *Proc. 20th USENIX Conf. Secur.*, USA, 2011, pp. 1–16.

[13] Q. Zhang, H. Sun, X. Wu, and H. Zhong, "Edge video analytics for public safety: A review," *Proc. IEEE*, vol. 107, no. 8, pp. 1675–1696, Aug. 2019.

[14] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[15] H. Zhong, Y. Zhou, Q. Zhang, Y. Xu, and J. Cui, "An efficient and outsourcing-supported attribute-based access control scheme for edge-enabled smart healthcare," *Future Gener. Comput. Syst.*, vol. 115, pp. 1330–1334, 2020.

[16] K. Fan, T. Liu, K. Zhang, H. Li, and Y. Yang, "A secure and efficient outsourced computation on data sharing scheme for privacy computing," *J. Parallel Distrib. Comput.*, vol. 135, pp. 169–176, 2020.

[17] H. Qian, J. Li, Y. Zhang, and J. Han, "Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation," *Int. J. Inf. Secur.*, vol. 14, no. 6, pp. 487–497, 2015.

[18] Y. Miao, Q. Tong, K.-K. R. Choo, X. Liu, R. H. Deng, and H. Li, "Secure online/offline data sharing framework for cloud-assisted industrial Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8681–8691, Oct. 2019.

[19] Y. Yu, L. Xue, Y. Li, X. Du, M. Guizani, and B. Yang, "Assured data deletion with fine-grained access control for fog-based industrial applications," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4538–4547, Oct. 2018.

[20] Y. Yang, X. Zheng, and C. Tang, "Lightweight distributed secure data management system for health Internet of Things," *J. Netw. Comput. Appl.*, vol. 89, pp. 26–37, 2017.

[21] Y. Yang, X. Liu, and R. H. Deng, "Lightweight break-glass access control system for healthcare Internet-of-Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 1330–1334, Aug. 2018.

[22] X. Yan, H. Ni, Y. Liu, and D. Han, "Privacy-preserving multi-authority attribute-based encryption with dynamic policy updating in PHR," *Comput. Sci. Inf. Syst.*, vol. 16, no. 3, pp. 831–847, 2019.

[23] K. Fan, H. Xu, L. Gao, H. Li, and Y. Yang, "Efficient and privacy preserving access control scheme for fog-enabled IoT," *Future Gener. Comput. Syst.*, vol. 99, pp. 134–142, 2019.

[24] F. Guo, Y. Mu, and Z. Chen, "Identity-based online/offline encryption," in *Proc. Int. Conf. Financial Cryptography Data Secur.*, 2008, pp. 1330–1334.

[25] Y. Zhang, D. Zheng, Q. Li, J. Li, and H. Li, "Online/offline unbounded multi-authority attribute-based encryption for data sharing in mobile cloud computing," *Secur. Commun. Netw.*, vol. 9, no. 16, pp. 3688–3702, 2016.

[26] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 8, pp. 1343–1354, Aug. 2013.

[27] B. Qin, R. H. Deng, S. Liu, and S. Ma, "Attribute-based encryption with efficient verifiable outsourced decryption," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 7, pp. 1384–1393, Jul. 2015.

[28] T. V. X. Phuong, G. Yang, and W. Susilo, "Hidden ciphertext policy attribute-based encryption under standard assumptions," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 1, pp. 35–45, Jan. 2016.

[29] M. Lyu, X. Li, and H. Li, "Efficient, verifiable and privacy preserving decentralized attribute-based encryption for mobile cloud computing," in *Proc. IEEE 2nd Int. Conf. Data Sci. Cyberspace*, 2017, pp. 1330–1334.

[30] H. Ma, E. Dong, Z. Liu, and L. Zhang, "Privacy-preserving multi-authority ciphertext-policy attribute-based encryption with revocation," in *Proc. Int. Conf. Broadband Wireless Comput., Commun. Appl.*, 2017, pp. 1330–1334.

[31] H. Zhong, W. Zhu, Y. Xu, and J. Cui, "Multi-authority attribute-based encryption access control scheme with policy hidden for cloud storage," *Soft Comput.*, vol. 22, no. 1, pp. 243–251, 2018.

[32] R. Canetti, H. Krawczyk, and J. B. Nielsen, "Relaxing chosen-ciphertext security," in *Proc. Annu. Int. Cryptol. Conf.*, 2003, pp. 1330–1334.

[33] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2011, pp. 1330–1334.

[34] A. Kate, G. Zaverucha, and I. Goldberg, "Pairing-based onion routing," in *Proc. Int. Workshop Privacy Enhancing Technol.*, 2007, pp. 1330–1334.

[35] J. Shao, Y. Zhu, and Q. Ji, "Privacy-preserving online/offline and outsourced multi-authority attribute-based encryption," in *Proc. IEEE/ACIS 16th Int. Conf. Comput. Inf. Sci.*, 2017, pp. 1330–1334.

[36] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 1330–1334.