Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.059969

ARTICLE

Weighted Attribute Based Conditional Proxy Re-Encryption in the Cloud

Xixi Yan[1], **Jing Zhang**[2,*] **and Pengyu Cheng**[2]

[1]Henan Key Laboratory of Network Cryptography Technology, Network Information Security and Cryptography Laboratory, School of Software, Henan Polytechnic University, Jiaozuo, 454003, China

[2]School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, 454000, China

*Corresponding Author: Jing Zhang. Email: zhangj202227@163.com

Received: 21 October 2024; Accepted: 21 January 2025; Published: 26 March 2025

ABSTRACT: Conditional proxy re-encryption (CPRE) is an effective cryptographic primitive language that enhances the access control mechanism and makes the delegation of decryption permissions more granular, but most of the attribute-based conditional proxy re-encryption (AB-CPRE) schemes proposed so far do not take into account the importance of user attributes. A weighted attribute-based conditional proxy re-encryption (WAB-CPRE) scheme is thus designed to provide more precise decryption rights delegation. By introducing the concept of weight attributes, the quantity of system attributes managed by the server is reduced greatly. At the same time, a weighted tree structure is constructed to simplify the expression of access structure effectively. With conditional proxy re-encryption, large amounts of data and complex computations are outsourced to cloud servers, so the data owner (DO) can revoke the user's decryption rights directly with minimal costs. The scheme proposed achieves security against chosen plaintext attacks (CPA). Experimental simulation results demonstrated that the decryption time is within 6–9 ms, and it has a significant reduction in communication and computation cost on the user side with better functionality compared to other related schemes, which enables users to access cloud data on devices with limited resources.

KEYWORDS: Cloud service; conditional proxy re-encryption; user revocation; weighted attribute

1 Introduction

Proxy re-encryption (PRE) refers to a cryptographic system with secure ciphertext transformation, where the agent acts as an intermediary for the ciphertext transformation, allowing the user to obtain the intended ciphertext. At the same time, in the process of ciphertext transformation, the agent can neither obtain the plaintext information nor the encryption key of the DO, so the agent can carry out ciphertext transformation in an untrustworthy environment. However, the disadvantage of the traditional PRE is that if the agent has the correct key for the DO, the agent can change all the ciphertexts of the DO to be decrypted by the user directly. The more granular delegation of decryption permissions that are required in a lot of application scenarios is not satisfied by this.

It was not until 2009 that Weng et al. [1] introduced the idea of Conditional proxy re-encryption (CPRE) to address this issue. This approach allows the DO to define a condition during the encryption process, which is then used to construct ciphertext with a transformation condition that can only be satisfied if the agent has a re-encryption key that satisfies that condition. The introduction of CPRE limits the ability of the agent to re-encrypt data, limiting it to transforming the partially satisfied ciphertexts of the DO even when the agent possesses the re-encryption key. Subsequently, various attribute-based conditional proxy re-encryption (AB-CPRE) schemes emerged to restrict the user's decryption rights further. In the AB-CPRE scheme, the access

structure will be used as a condition to restrict the agent from transforming the ciphertext. This optimizes the CPRE scheme in conjunction with the granular access control structure of attribute-based encryption (ABE), which provides a more granular delegation of decryption permissions than before.

The AB-CPRE scheme currently in use assumes that two users with identical attributes have the same access rights. For instance, an access policy should be set so that only physicians over 45 years old can access encrypted files. However, the physician attribute may comprise 'director physician', 'associate director physician', 'house physician', 'doctor in charge of a case'. If we were to allow the director physician, associate director physician, or house physician over 45 years old to access the file, we would have to add three of the four sub-attributes divided under the physician attribute to the access structure. This access policy is {{"Director Physician" OR "Associate Director Physician" OR "house physician"} AND "Age > 45"}. Considering all possible situations will inevitably lead to a greater number of characteristics in the management system, making the access structure more complicated. The current AB-CPRE adopts the same access structure as its predecessor, treating attribute characteristics at the same level without hierarchical classification based on importance.

According to the needs of the actual situation, each attribute of the user is given a weight according to its own importance so that the user gets a set of weighted attributes. Assign weights of 4, 3, 2, and 1 to 'director physician', 'associate director physician', 'house physician', 'doctor in charge of a case', respectively. If a weighted access policy is constructed, it can only satisfy the above three cases. This weighted access policy is {"Physician {2}" AND "Age > 45"}, where "Physician {2}" indicates that the node requirement can only be satisfied if the user's attribute weight is not less than 2. As shown in Fig. 1, the improved weighted access tree is simpler, and the effect is more pronounced as the user attributes are divided into more levels.

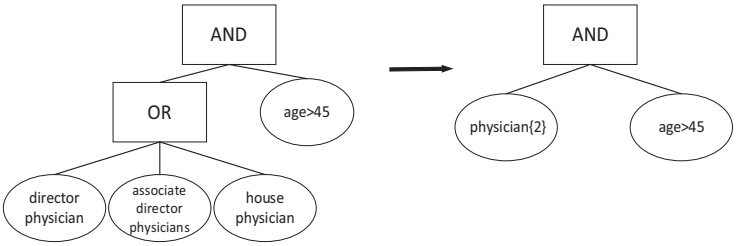

Figure 1: Comparison of access policy

By assigning appropriate weights to the attributes, the identity characteristics of each user are further refined, and the agent can more accurately restrict the user's decryption rights through the weighted access policy. At the same time, setting a weighted access policy can further simplify the access structure. Therefore, in order to granular delegation of decryption permissions for AB-CPRE, it is meaningful to propose a weighted attribute-based conditional proxy re-encryption (WAB-CPRE) scheme at this stage.

1.1 Related Work

Blaze et al. [2] first presented the idea of PRE, who created a scheme that was fundamentally comparable to the ElGamal encryption method [3] by using a public-key cryptography algorithm with straightforward proxies. As PRE is slowly becoming an important cryptographic primitive in secure cloud computing and cloud storage, PRE is required to support a more granular delegation of decryption permissions for practical applications at this stage. Weng et al. [1] presented the idea of CPRE for the first time and proved that the scheme satisfies CCA security. However, early CPRE schemes used simple keywords or "AND" joins of

multiple keywords as conditions for re-encryption by the agent, but these CPRE schemes are still somewhat unsatisfactory for the granular delegation of decryption permissions required by practical examples.

An AB-CPRE system was initially presented by Zhao et al. [4] in 2010. It allows attribute-based management of decryption delegation by allowing access structures and attribute sets to be established. The scheme embeds access structure in the re-encryption key, the collection of attributes into the ciphertext, for the first time realizing granular conditions than the keyword. A new AB-CPRE technique was suggested by Yang et al. [5], who also showed how useful the scheme is for cloud deployment. This scheme uses a ciphertext policy to realize AB-CPRE, which is more natural and flexible compared to the literature [4]. An identity-based CPRE will be presented by Yao et al. [6] in 2021. It solves the problem that the traditional PRE scheme focuses on access authorization and ignores key updates and ciphertext evolution. Subsequently, a granular certificateless CPRE scheme without pairings was proposed by Yang et al. [7]. It not only solves the problem of the traditional certificates CPRE scheme that does not support one-to-many transformations but also improves the efficiency of the scheme through the without pairings approach.

In 2023, Tang et al. [8] proposed an attribute-based verifiable CPRE scheme, which is based on homomorphic signatures, and introduced a verification server so that the re-encrypted ciphertext can be verified by a verification server. The scheme can determine whether the original ciphertext has been correctly transformed and thus can effectively detect the agent's illegal activities. First, in 2024, a single-hop directly revocable AB-CPRE was proposed based on standard LWE assumptions [9], which provides more flexible access control while allowing the delegate to dynamically authorize and revoke a user's decryption privileges.

However, none of the existing AB-CPRE schemes describe the importance of attributes, but in reality, the access rights of different users with the same attribute may not be the same. Liu et al. [10] generalized the traditional ABE and proposed a weighted ABE scheme by performing weighting operations on attributes according to their importance. Only when the ciphertext contains a set of attributes with different weights that satisfy the access structure can the user decrypt the ciphertext. However, this scheme is only an improvement of the traditional ABE, and there is no function for a semi-trusted agent to securely transfer the decryption right of its ciphertext. Therefore, this scheme, compared to the scheme [10], in addition to supporting the delegation of decryption rights, also has a more fine-grained delegation of decryption rights than the current AB-CPRE. Meanwhile, this program supports the direct revocation of user access rights, which solves the complex revocation work of traditional attribute-based encryption. High user-side efficiency is also a major advantage of this scheme, which enables users to decrypt access to data on cloud services on resource-constrained devices.

1.2 Contribution

A WAB-CPRE scheme will be presented in conjunction with Liu et al. [10] approach to achieve a more flexible and granular delegation of decryption permissions in a cloud computing environment. The following are the principal contributions:

- By introducing the concept of weighted attributes, the user's identity characteristics are refined, and the decryption delegation of the AB-CPRE scheme is fine-grained. Assigning weights to each user's attributes according to the degree of importance can relieve the number of system attributes and ease the server management burden. The effect of this simplification is more pronounced when the user hierarchizes more attributes or has more levels at which each attribute is hierarchized.
- In conjunction with CPRE technology, a weighted access tree will be set up to control user access rights. The cloud service provider (CSP) can successfully transform the ciphertext decryption permission to the user as long as the user's collection of weighted attributes satisfies the weighted access structure set by the DO.

- By assigning the re-encryption procedures to the server, outsourcing techniques help the DO cut down on computational and communication expenses. At the same time, the scheme supports the direct revocation of the user's decryption right, eliminating the tedious procedure of updating the ciphertext data and distributing re-encryption keys in the original revocation function.
- The security of the CPA is accomplished under decision l-Expanded Bilinear Diffie-Hellman exponent (l-Expanded DBDH) assumptions. To illustrate the greater user-side computing and communication efficiency of the proposed method, simulation experiments are carried out in addition to a comparison and analysis of its functionality and performance.

1.3 Organization

The remainder of the document is arranged as follows: Section 2 focuses on basic preparatory knowledge. Section 3 gives the algorithm definition and security model, and the application scenarios are specifically analyzed at the end of this section. Section 4 details the specific construction and security analysis. Then, in Section 4, the performance analysis is made by comparing the scheme with the existing schemes. Lastly, Section 6 provides the paper's conclusion.

2 Preliminaries

2.1 Access Structure

Definition 1: If there are n participants, let $\{P_1, P_2, \ldots, P_n\}$ be a collection of participants. The set $\mathbb{P} \subseteq 2^{\{P_1, P_2, \cdots, P_n\}}$ is monotone if $\forall D, E$: $D \in \mathbb{P}$, $D \subseteq E$ then $E \in \mathbb{P}$, where the access structure is a nonempty subset of $\{P_1, P_2, \ldots, P_n\}$, $\mathbb{P} \subseteq 2^{\{P_1, P_2, \cdots, P_n\}} \setminus (\varnothing)$. Assume that there is an authorized subset of set $\mathbb{P}$ and that there are illegal sets for the other sets that are not in $\mathbb{P}$.

2.2 Weighted Access Tree

In the WAB-CPRE scheme, the DO uses a collection of descriptive weighted attributes $\mathbb{A}$ to structure the associated re-encryption key. DO specifies the desired $\mathbb{P}$ for the message and creates the ciphertext under $\mathbb{P}$. When a collection of descriptive weighted attributes $\mathbb{A}$ satisfies $\mathbb{P}$, the CSP successfully transforms the ciphertext and transmits it to the user. We name Γ, which contains both leaf and non-leaf nodes, as its weighted access tree. The non-leaf node x consists of the number of sub-nodes num_x and the threshold k_x, where $0 \leq k_x \leq num_x$. When $k_x = num_x$ and $k_x = 1$ are set to "AND" and "OR" gates, the leaf node represents a weighted attribute, and the k_x of its parent node is increased by 1 if the user's weight for that attribute is satisfied. The function $parent(x)$ can be called on the parent of node x, while the function $att(x)$ can extract the weighted attribute connected to the leaf node. Assign a unique index value to each child node of a non-leaf node in order from 1 to num_x, and the function $index(x)$ is used to represent the index value of that child node.

2.3 Satisfying a Weighted Access Tree

Γ_n is defined as a weighted access tree with root x. Define a collection of attributes $\mathbb{A}$ satisfies Γ_x as $\Gamma_x(\mathbb{A}) = 1$. We define a recursive algorithm to compute the value of $\Gamma_x(\mathbb{A})$. The Γ_x of a non-leaf node x returns 1, and a specific number of its leaf nodes must be required to all have a $\Gamma_{x'}$ of 1. If x is a leaf node, the value of $\Gamma_{x'}$ is 1 when the weight of the attribute attribute λ_x in $\mathbb{A}$ is not less than the weight of the attribute of the leaf node, that is, $weight(\lambda_x) \geq weight(att(x))$.

2.4 The Decisional l-Expanded BDHE Assumption

We introduce the paper [11] constructing the decision l^*-Expanded Bilinear Diffie-Hellman Exponent problem. The $a, b, c_0, \cdots, c_{l+1}, d \in Z_P$ chosen randomly and g is a generating element of the cyclic group G. Assume that an opponent is provided $X =$

$$g, g^a, g^b, g^{ab/d}, g^{b/d}$$
$$\forall_{i\in[0,2l+1],i\neq l+1,j\in[0,l+1]}\, g^{a^i s}, g^{a^i bs/c_j}$$
$$\forall_{i\in[0,l+1]}\, g^{a^i b/c_i}, g^{c_i}, g^{a^i d}, g^{abc_i/d}, g^{bc_i/d}$$
$$\forall_{i\in[0,2l+1],j\in[0,l+1]}\, g^{a^i bd/c_j}$$
$$\forall_{i\in[0,l+1],i\neq j} g^{a^i bc_j/c_i}$$

Define random tuple $(\mathbf{X}, R) \in G_T$ and tuple $\left(X, T = e\left(g,g\right)^{a^{l+1}bs}\right) \in G_T$, where R is chosen independently from G_T. Let Algorithm B have an advantage in solving the decision l-Expanded BDHE in G is $\boldsymbol{\varepsilon}$.

$$\left|\Pr\left[\boldsymbol{\mathcal{B}}\left(X, T = e\left(g,g\right)^{a^{l+1}bs}\right) = 0\right] - \Pr\left[\boldsymbol{\mathcal{B}}\left(X, T = R\right) = 0\right]\right| \geq \boldsymbol{\varepsilon}$$

Definition 2: The assumption of decision l-Expanded BDHE is valid if no algorithm is able to obtain a non-negligible advantage in polynomial time when addressing the problem.

3 Definition and Application

3.1 Algorithm Definition

$Setup\left(1^\lambda, W\right) \rightarrow Params$: After inputting the master entity's security parameters 1^λ and attribute universe W, this algorithm produces the scheme's public parameters, and the public parameters are published by the master entity.

$KenGen\left(u\right) \rightarrow Sk_u, Pk_u$: The algorithm outputs the key pairs Sk_u, Pk_u for the user u.

$Enc\left(Pk_A, m, \Gamma\right) \rightarrow C_A$: Given an access tree Γ input the message m that you want to encrypt and the DO's public key Pk_A. The algorithm generates ciphertext C_A.

$ReKenGen\left(Sk_A, \mathbb{A}_B, Pk_B\right) \rightarrow Sk_{A\rightarrow B}$: After inputting the private key Sk_A of the DO and a collection of weighted attributes $\mathbb{A}_B$ and public key Pk_B of the user B. Subsequently, the algorithm outputs $Sk_{A\rightarrow B}$ associated with the set of weighted attributes.

$ReEnc\left(Sk_{A\rightarrow B}, C_A\right) \rightarrow C_B$: The algorithm is performed by the CSP. Enter the ciphertext C_A and the user's re-encryption key $Sk_{A\rightarrow B}$. If the collection of weighted attributes $\mathbb{A}_B$ satisfies the weighted access structure $\mathbb{P}$, the algorithm sends the ciphertext C_B to the user.

$Dec\left(Sk_B, C_B\right) \rightarrow m$: The decryption algorithm takes the key Sk_B and the converted ciphertext C_B as input, which produces the plaintext m if Sk_B is a working decryption key.

3.2 Security Model Definition

Init. The adversary $\mathcal{A}$ chooses and declares a weighted access structure $\mathbb{P}^*$.

Setup. Challenger $\mathcal{B}$ executes the $Setup$ algorithm and outputs $params$ and then performs the $KenGen$ algorithm to obtain the key pairs Pk_Q/Sk_Q. Finally, $\mathcal{A}$ sends the output $params$ to $\mathcal{B}$.

Phase 1. Using a different collection of weight attributes $A_1, A_2, A_3, \ldots, A_{q_1}$, respectively, adversary $\mathcal{A}$ asks challenger $\mathcal{B}$ for the re-encryption keys. Subsequently, challenger $\mathcal{B}$ responds by executing the $KenGen$ algorithm to distribute the key pairs Pk_J/Sk_J a for adversary $\mathcal{A}$ and by using the $ReKenGen\left(Sk_Q, \mathbb{A}_j, Pk_Q\right) \rightarrow rk_{Q\rightarrow J}$ algorithm to create a re-encryption key for $\mathcal{A}$. Finally, the key pairs Pk_J/Sk_J and rk are returned to the adversary.

Challenge. Two challenge messages of identical length, m_1 and m_2, are sent by the adversary. The challenger $\mathcal{B}$ tosses a coin β at random and encrypts the message m_β by running an Enc algorithm under a weighted access structure $\mathbb{P}^*$ and submits the ciphertext C^* to $\mathcal{A}$.

Phase 2. Using a different collection of weight attributes $A_{q_1+1}, A_{q_1+2}, A_{q_1+3}, \ldots, A_{q_2}$ again, adversary $\mathcal{A}$ asks the challenger for the re-encryption keys. As in phase 1, the challenger returns Pk_J and rk to the adversary if A_i satisfies $\mathbb{P}^*$; otherwise, Pk_J/Sk_J and rk are returned to the adversary.

Guess. A guess about β is output by the adversary.

Definition 3: A WAB-CPRE scheme satisfies CPA security if, for any PPT adversary $\mathcal{A}$, there exists a negligible function $\boldsymbol{\varepsilon}(\boldsymbol{\mathcal{K}})$ such that $Adv_{\boldsymbol{\mathcal{A}}} \leq \boldsymbol{\varepsilon}(\boldsymbol{\mathcal{K}})$, which defines $\mathcal{A}$'s advantage in this game to be $Adv_{\boldsymbol{\mathcal{A}}} = \mathbf{Pr}[\boldsymbol{\beta}' = \boldsymbol{\beta}] - \frac{1}{2}$.

3.3 Application Scenario

The system model of the WAB-CPRE scheme includes four entities: the data owner (DO), the user, the cloud service provider (CSP), and the key generation center (KGC). Using the hospital employee system as an example, we can better demonstrate the system model by designating the DO Alice and the user Bob as the medical staff and the hospital administration, respectively. Assuming that the collection of weighted attributes of the healthcare worker meets the weighted access structure set by the hospital administrator, Fig. 2a shows the data sharing procedure from the hospital administrator (Alice) to the healthcare worker (Bob), and Fig. 2b illustrates the multiple operations involved between the various entities.

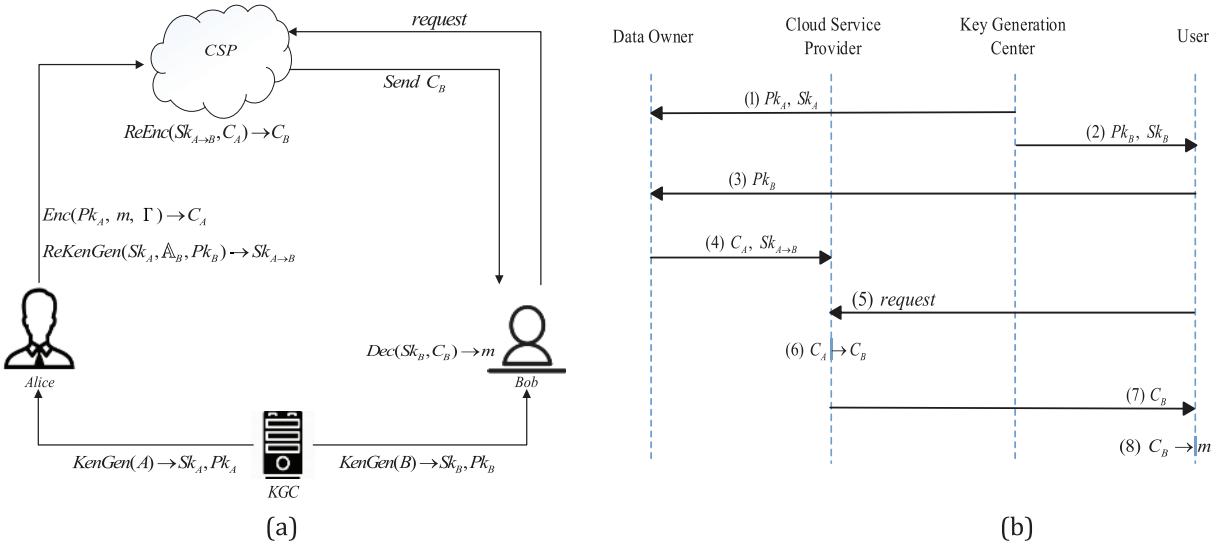

Figure 2: System model and workflow

First, Alice sets the access rights for each file according to the roles of the accessors and delivers the encrypted ciphertext to the CSP. Subsequently, Alice assigns re-encryption keys to healthcare workers (which

includes Bob) at different position levels in each department and uploads them to the CSP. The CSP stores a large amount of received ciphertext data and keys in two separate tables. Bob notifies the CSP of his requirement for access to the pertinent files by sending a request. The CSP extracts the appropriate ciphertext file and Bob's re-encryption key from a lookup table and performs the $ReEnc$ algorithm. If Bob's collection of attributes meets the weighted access structure, the CSP successfully transforms the ciphertext and transmits it to Bob. At last, Bob runs the algorithm to decrypt and extract the plaintext data.

The system takes advantage of cloud services by outsourcing large amounts of data and complex calculations to a CSP, which acts as a semi-trusted platform and will not extract any information from the uploaded data. Having the correct re-encryption key is an important condition for CSP to securely translate decryption permissions. When a healthcare worker leaves the hospital or joins the staff, there is no revocation of the role attribute in the system, but rather the revocation of the old healthcare worker's access rights and the granting of the new healthcare worker's access rights. Therefore, when a healthcare worker leaves and joins the staff, the system can remove or add the re-encryption key for that healthcare worker directly from the cloud.

4 Construction and Security Analysis

4.1 The Construction

$Setup\left(1^{\lambda}, W\right)$: First, the security parameter 1^{λ} and the attribute domain W are input for the algorithm. Define a bilinear map $e: G_0 \times G_0 \rightarrow G_T$, where G_0 and G_T are cyclic groups of prime order p, and g is chosen as the generators of G_0, a hash function $H: \{0,1\} \rightarrow G_0$. The scheme assigns a maximum weight of l_i to each attribute separately. The random number $z_i, h_{(i0)}, h_{(i1)}, \ldots, h_{(il_i)} \in G_0$ is generated for $\forall w_i \in W$ and published (Table 1 shows the meaning of each symbol in this scheme):

$$Params = \{e(g,g), g, H(\cdot), {}_{\forall w_i \in W}\{z_i, h_{(i0)}, \ldots, h_{(il_i)}\}\}$$

Table 1: Notations used

Notation	Description
W	Attribute domain
p	Big prime numbers
$\mathbb{Z}_p$	Multiplicative groups of integers of order p
G_T	Cyclic groups of order T
g	Generator of a sub-group of G_T
l_i	Attribute i maximum weight
ω_i	User attribute i weight
u	The identity of user
k_x	Node x threshold

$KenGen\,(u)$: The key generation center first selects $x_u, \alpha_u, \beta_u \in \mathbb{Z}_p$ for user u randomly and computes $y_u = g^{x_u}, h_u = g^{\beta_u}, f_u = e\,(g,g)^{\alpha_u}$, then outputs user's key pairs Pk_u, Sk_u.

$Sk_u = \{x_u, \alpha_u, \beta_u\}, Pk_u = \{y_u, h_u, f_u\}$.

$Enc\,(Pk_A, m, \Gamma)$: The DO inputs the public key Pk_A, the message m, and the weighted access tree Γ. Let Y be the collection of leaf nodes of a weighted access tree Γ. Each leaf node λ_i represents an attribute that has a weight value of ω_i, where $0 < \omega_i < l_i$. For $\forall \lambda_i \in Y$, the encryption algorithm randomly generates

$s_{(i0)}, s_{(i1)}, \ldots, s_{(i\omega_i-1)} \in Z_P$. Generates the secret value $s \in Z_P$ of the root node R and generates a polynomial q_R of order $k_R - 1$ such that $q_R(0) = s$. Then, a $k_x - 1$ order polynomial is generated for all child nodes x of the root node R, setting $q_x(0) = q_{parent(x)}(index(x))$ finally until all leaf nodes are assigned the unique secret value $s_{(i\omega_i)}$. The function $att(x)$, which indicates the attributes connected to that leaf node x, is defined exclusively in cases when x is a leaf node.

First, set $c = m \cdot {f_A}^s$ and $c' = {h_A}^s$.

Then let $0 \le j \le \omega_i$, when $j = 0$, for $\forall \lambda_i \in Y$ set:

$$c_{(i0)} = g^{s_{(i0)}}, \tilde{c}_{(i0)} = \left(h_{(i0)}\right)^{s_{(i0)}}$$

when $0 < j < \omega_i$, for $\forall \lambda_i \in Y$ set:

$$c_{(ij)} = g^{s_{(ij)}}, \tilde{c}_{(ij)} = \left(h_{(ij)}\right)^{s_{(ij)}} \cdot H\left(att\left(\lambda_i\right)\right)^{s(ij-1)}$$

when $j = \omega_i$, for $\forall \lambda_i \in Y$ set:

$$c_{(ij)} = g^{q_{\lambda_i}(0)}, \tilde{c}_{(ij)} = \left(h_{(ij)}\right)^{q_{\lambda_i}(0)} \cdot H\left(att\left(\lambda_i\right)\right)^{s(ij-1)}$$

Finally, output ciphertext:

$$C_A = \left\{c, c', {}_{\forall \lambda_i \in Y}\left\{c_{(ij)}, \tilde{c}_{(ij)}\right\}_{j \in [0,\omega_i]}\right\}$$

$ReKenGen\left(Sk_A, \mathbb{A}_B, Pk_B\right)$: The algorithm starts by picking a random number $r \in Z_p$ for $\forall w_i \in \mathbb{A}_B$ pick $r_{(ij)} \in Z_p$, where $j \in [0, \omega_i']$ (ω_i' denotes the magnitude of the weight of the weighted attribute w_i).

Firstly, set $k = \left({y_B}^{\alpha_A} g^r\right)^{-\beta_A}$

Secondly, for $\forall w_i \in \mathbb{A}_B$, when $j = 0$ set:

$$k_{(i0)} = g^r \left(h_{(i0)}\right)^{r_{(i0)}}, \tilde{k}_{(i0)} = g^{r_{(i0)}}$$

when $j \in (0, \omega_i']$ set:

$$k_{(ij)} = g^{-r} H\left(w_i\right)^{r_{(ij)}}, \tilde{k}_{(ij)} = g^{r_{(ij)}}, \tilde{\tilde{k}}_{(ij)} = g^r \left(h_{(ij)}\right)^{r_{(ij)}}$$

Finally, output the re-encryption key:

$$Sk_{A\to B} = \left\{k, k_{(i0)}, \tilde{k}_{(i0)}, {}_{\forall w_i \in P}\left\{k_{(ij)}, \tilde{k}_{(ij)}, \tilde{\tilde{k}}_{(ij)}\right\}_{j \in \left(0,\omega_i'\right]}\right\}$$

$ReEnc\left(Sk_{A\to B}, C_A\right)$: First, define a recursive algorithm $ReEncNd_n\left(Sk_{A\to B}, C_A\right)$, where n is a node of the tree Γ. ① In the case where n is a leaf node, then define $z = att(n)$ and give the following definition: If $z \notin \mathbb{A}_B$, then $ReEncNd_n\left(Sk_{A\to B}, C_A\right) = \perp$; otherwise, let $ReEncNd_n\left(Sk_{A\to B}, C_A\right) = F_n$. Then, run the recursive algorithm $ReEncNd_n\left(Sk_{A\to B}, C_A\right)$ to find the specific value of F_n.

First, when $j = 0$ set:

$$B_{(i0)} = e\left(c_{(i0)}, k_{(i0)}\right) \cdot e\left(\tilde{c}_{(i0)}, \tilde{k}_{(i0)}\right)^{-1} = e(g,g)^{r \cdot s_{(i0)}}$$

Second, when $j \in (0, \omega_i]$ set:

$$\begin{aligned} B_{(ij)} &= B_{(ij-1)} \cdot e\left(c_{(ij-1)}, k_{(ij)}\right) \cdot e\left(\tilde{c}_{(ij)}, \tilde{k}_{(ij)}\right)^{-1} \cdot e\left(c_{(ij)}, \tilde{\tilde{k}}_{(ij)}\right) \\ &= e(g,g)^{rs_{(ij)}} \end{aligned}$$

Finally, if $\omega_i \le \omega_i{}'$ is satisfied, then $B_{(i\omega_i)} = e(g,g)^{r \cdot q_{\lambda_i}(0)}$ and let $F_n = B_{(i\omega_i)} = e(g,g)^{r \cdot q_n(0)}$.

② In the case where n is a non-leaf node, the recursive algorithm $ReEncNd_x(Sk_{A\to B}, C_A)$ is called. Let S_n be the set of F_v of size not less than t_n, where v is a leaf node of n and $F_v \neq \perp$. Then $F_n = \perp$ if there isn't such a collection, otherwise:

$$\begin{aligned} F_n &= \prod_{v \in S_n} F_v{}^{\triangle_{i,S_n'}(0)}, where^{i=index(v),}_{S_n'=\{index(v):v\in S_n\}} \\ &= \prod_{v \in S_n} \left(e(g,g)^{r \cdot q_v(0)}\right)^{\triangle_{i,S_n{}'}(0)} \\ &= \prod_{v \in S_n} \left(e(g,g)^{r \cdot q_{parent(v)}(index(v))}\right)^{\triangle_{i,S_n{}'}(0)} \\ &= \prod_{v \in S_n} \left(e(g,g)^{r \cdot q_n(i)}\right)^{\triangle_{i,S_n{}'}(0)} \\ &= e(g,g)^{r \cdot q_n(0)} \end{aligned}$$

③ If n is the root node, then $F_n = e(g,g)^{r \cdot s}$.

At last compute $\tilde{c} = \frac{e(k,c')}{e(g,g)^{r \cdot s}} = f_A^{s \cdot x_B}$ and $c = m \cdot {f_A}^s$, then outputs the re-encrypted ciphertext as $C_B = \{c, \tilde{c}\}$.

$Dec(Sk_B, C_B)$: The user performs the decryption algorithm. After entering a valid private key Sk_B and the re-encrypted ciphertext C_B, the following computation is performed to obtain the plaintext message:

$$\frac{c}{(\tilde{c})^{x_B{}^{-1}}} = \frac{m \cdot {f_A}^s}{\left({f_A}^{s \cdot x_B}\right)^{x_B{}^{-1}}} = m$$

4.2 Security Analysis

There are two separate situations to take into account. ① The first case is when none of the collection of attributes A_j satisfies the $\mathbb{P}^*$. Adversary $\mathcal{A}$ would receive the key pairs Pk_J/Sk_J and the re-encryption key $rk_{Q\to J}$ that challenger $\mathcal{B}$ provided to the user. ② The second case is when at least one of the attribute collections A_j satisfies the weighted access structure $\mathbb{P}^*$. Therefore, the challenger $\mathcal{B}$ returns the user's public keys Pk_J and $rk_{Q\to J}$ to $\mathcal{A}$.

Assume, for the first example, that there is a polynomial-time $\mathcal{A}$ whose advantage in attacking the selective access tree model in this scheme is $Adv_{\mathcal{A}}$. The security model for this case is the same as in the paper [10]. The following describes in detail the parameterization, ciphertext generation, and re-encryption key request of adversary $\mathcal{A}$ by simulator $\mathcal{B}$ during the simulation process. The simulation process is shown below:

Definition 4: Assume that the decisional l^*-Expanded BDHE assumption is valid. Then, given a challenge access tree Γ^*, no adversary with polynomial time could selectively compromise our system.

First, challenger $\mathcal{B}$ chooses two cyclic groups, G_0 and G_T, where g is the generator of G_0. $\mathcal{B}$ flips an arbitrary binary coin μ. When $\mu = 0$, $\mathcal{A}$ sets $\left(\mathbf{X}, T = e\left(g, g\right)^{a^{\omega^*+1}bs}\right)$; when $\mu = 1$, the challenger $\mathcal{B}$ sets $(\mathbf{X}, R)$, where R is chosen independently from G_T.

Init. $\mathcal{A}$ selects and declares a weighted access structure $\mathbb{P}^*$.

Setup. Choose random indices $v_{(k0)} \cdots v_{(k\omega^*)}, v_{z_k}, a, c_j, b, d, r \in Z_p$, and ω^* is the weight value of attribute k, $k \in \boldsymbol{\mathcal{U}}$ ($\boldsymbol{\mathcal{U}}$ is attribute universe). The challenger uses an empty table in order to mimic the hash function H. When $\mathcal{A}$ asks a query k that has never been asked before, $\mathcal{B}$ computes $g^{v_{z_k}}$ and adds a table entry $(k, g^{v_{z_k}})$ to the table, returning $g^{v_{z_k}}$ to $\mathcal{A}$. The challenger chooses a random identity Q and executes the $KenGen$ algorithm to obtain key pairs Pk_Q/Sk_Q, where $e\left(g, g\right)^r = e\left(g^a, g^b\right)$ is defined and parameterized:

$$h_{(k0)} = g^{v_{(k0)}} \cdot \prod_{j\in[1\omega^*]} g^{-a^j b/c_j}, h_{(\mathrm{ki})} = g^{v_{(\mathrm{ki})}} \cdot \prod_{j\in[0\omega^*+1], i\neq j} g^{-a^{\left(\omega^*+1-j\right)} b/c_{\left(\omega^*+1-j\right)}}.$$

Phase 1. Adversary $\mathcal{A}$ makes a key query to the challenger. $\mathcal{B}$ implicitly setting $r_{(k0)} = \sum_{i\in S_{(k0)}} g^{c_{i+1}}$ and executes the $KenGen$ algorithm using a random user identity and generates the key pairs Pk_J/Sk_J for adversary $\mathcal{A}$. For $\forall k \in S^*$, define a set $S_{(kt)}$, where $t \in [0, \omega^*]$.

When $t = 0$, set $\tilde{K}_{(k0)} = g^{r_{(k0)}} = \prod_{i\in S_{(k0)}} g^{c_{i+1}}$

$$\begin{aligned} K_{(k0)} &= g^{ab}\left(h_{(k0)}\right)^{r_{(k0)}} \\ &= g^{ab} \cdot \left(\tilde{K}_{(k0)}\right)^{v_{(k0)}} \cdot \prod_{\substack{j\in[1\omega^*], i\in S_{(k0)} \\ j\neq i+1}} g^{-a^j b c_{i+1}/c_j} \end{aligned}$$

When $t \in (0, \omega^*]$, set $\tilde{K}_{(k\mathbf{t})} = g^{r_{(k\mathbf{t})}} = \prod_{i\in S_{(k\mathbf{t})}} g^{a^i d}$

$$\begin{aligned} K_{(k\mathbf{t})} &= g^{-ab} \cdot \left(\tilde{K}_{(k\mathbf{t})}\right)^{v_{z_k}} \\ \tilde{\tilde{K}}_{(kt)} &= g^{ab}\left(h_{(kt)}\right)^{r_{(kt)}} \\ &= g^{ab}\left(\tilde{K}_{(kt)}\right)^{v_{z_k}} \cdot \prod_{\substack{j\in[0\omega^*+1], i\in S_{(kt)} \\ j\neq i}} g^{-a^{\omega^*+1-j+i} bd/c_{\omega^*+1-j}}. \end{aligned}$$

Challenge. When $\mathcal{B}$ starts to create a challenge ciphertext, it is necessary first to establish the individual node polynomials of the access tree Γ_x. If the collection S does not satisfy Γ_x, then $\Gamma_x(S) = 0$. To access all sub-nodes x' of the tree Γ_x, define $\lambda_{x'} \in Z_p$ and compute $\lambda_{x'} = q_x\left(index\left(x'\right)\right)$. Ultimately, polynomials for the remaining nodes are recursively created.

When x' is a sub-node of x that satisfies the collection of weighted attributes, the value of $q_x\left(index\left(x'\right)\right)$ can be calculated. The simulator subsequently runs the recursive algorithm $RecSat\left(\Gamma_{x'}, S, q_x\left(index\left(x'\right)\right)\right)$ to generate a polynomial for that node. If the node x' is an unsatisfiable child node, the value of $g^{q_x(index(x'))}$ can be computed by interpolation algorithm. The simulator then runs the recursive algorithm $RecUnsat(\Gamma_{x'}, S, g^{q_x(index(x'))})$ to generate a polynomial for the node. At last, the value of $q_k(0)$ for each leaf node is output, k.

Let $s_{(ki)} = a^i r'_k \in Z_p$. Two equal-length challenge messages m_1 and m_2, are submitted by adversary $\mathcal{A}$, and the challenger tosses a coin, β.

$$C_m = m_\beta \cdot T, c_{(k0)} = g^{r'_k},$$
$$\tilde{c}_{(k0)} = \left(h_{(k0)}\right)^{r'_k} = \left(g^{r'_k}\right)^{v_{(k0)}} \cdot \prod_{j \in \left[1, \omega_k^*\right]} g^{a^i b r'_k / c_j}.$$

When $i \in (0, \omega^*]$, it set:

$$c_{(ki)} = g^{a^i \cdot r'_k}, \tilde{c}_{(ki)} = \left(g^{a^i \cdot r'_k}\right)^{v_{(ki)}} \cdot \left(g^{a^{i-1} \cdot r'_k}\right)^{v_{z_k}} \cdot \prod_{j \in \left[1, \omega_k^*\right]} g^{a^{\omega^*+1-j+i} b r'_k / c_{\omega^*+1-j}}.$$

If $T = e(g,g)^{a^{\omega^*+1} bs}$ is a correct Expanded BDHE tuple, then C_m is the ciphertext encrypted by m_β. If not, then C_m is a random element in G_T, and $\mathcal{A}$ will not have access to valid information about β.

Guess. $\mathcal{A}$ will provide a guess β'. Judge the output according to the following rules: If $\beta' = \beta$, the simulator outputs 1. If not, the simulator outputs 0.

When the simulator outputs 1, the simulation attack view of adversary $\mathcal{A}$ is the same as the actual attack view, so that we can get $\Pr\left[\mathcal{B}\left(y, T = e(g,g)^{a^{\omega^*+1} bs}\right) = 0\right] = \frac{1}{2} + Adv_{\mathcal{A}}$. On the other hand, if the simulator returns 0, then the message m_β is concealed from $\mathcal{A}$, where T is a random selection from G_T. As a result, the simulator is able to play that game with a non-negligible advantage.

For the second scenario, the collection of weighted attributes A_j of the key query performed by $\mathcal{A}$ all satisfy the challenge weighted access structure $\mathbb{P}^*$. Thus, $\mathcal{A}$ can obtain the transformed ciphertext. A detailed description of the proposed scheme's security game under the CPA follows.

Init. $\mathcal{A}$ selects and declares a weighted access structure $\mathbb{P}^*$.

Setup. As in the same case, challenger $\mathcal{B}$ sets the public parameters and sends them to $\mathcal{A}$. Subsequently, $\mathcal{B}$ selects a random identity Q and executes the $KenGen$ algorithm to obtain the public Pk_Q and private key Sk_Q.

Phase 1. The adversary $\mathcal{A}$ performs a re-encryption key lookup operation by providing a weighted attribute set A_j to the challenger $\mathcal{B}$ (This weighted attribute set A_j satisfies the weighted access structure $\mathbb{P}^*$). For the weighted access structure A_j received by the challenger for the j th time, $\mathcal{B}$ first randomly chooses $x_J, \alpha_J, \beta_J \in \mathbb{Z}_p$ and generates the private key Sk_J of $\mathcal{A}$. Subsequently, select $r, r_{(it)}$, where $i \in A_j, t \in \left[0, \omega_i^*\right]$, and return $Pk_Q, Pk_J, rk = \left\{k = \left(y_J{}^{\alpha_Q} g^r\right)^{-\beta_Q}, k_{(it)}, \tilde{k}_{(it)}, \forall i \in A \left\{k_{(it)}, \tilde{k}_{(it)}, \tilde{\tilde{k}}_{(it)}\right\}_{t \in (0, \omega_i^*]}\right\}$ to $\mathcal{A}$.

Challenge. Challenger $\mathcal{B}$ chooses $s, s_{(it)} \in \mathbb{Z}_P$, where $i \in A_j, t \in \left[0, \omega_i^* - 1\right]$. Then, the challenger distributes secret values according to $\mathbb{P}^*$ declared by $\mathcal{A}$ so that each leaf node of Γ is assigned a unique secret value. Finally, after $\mathcal{A}$ sends two challenge messages m_1, m_2, $\mathcal{B}$ chooses $\vartheta \in Z_P$, and computes $c = e(g,g)^{\vartheta}, c' = g^{\beta \cdot s}$ to return the ciphertext C to $\mathcal{A}$.

Phase 2. The challenger keeps answering A's questions on re-encryption key generation.

Guess. A guess about β is output by the adversary $\mathcal{A}$.

We will cite a game [12] where the challenged ciphertext c is either $e(g,g)^{\alpha \cdot s}$ or $e(g,g)^{\vartheta}$, where $\vartheta \in Z_P$. The adversary $\mathcal{A}$ has to guess which is correct. Next, we prove with likelihood $1 - O\left(q^2/p\right)$ that A's guess

is $c = e(g,g)^{\alpha \cdot s}$ in the simulation. Adversary $\mathcal{A}$ has an advantage of at most $O\left(q^2/p\right)$ given $c = e(g,g)^{\alpha \cdot s}$, since the challenge message $c = e(g,g)^{\vartheta}$ is independent of the encrypted message.

When $\mathcal{A}$ requests the group oracles, we require $\mathcal{A}$ to obtain values only from the simulation. Here we consider an oracle query as a rational function $\pi = \gamma/\eta$ and $\mathcal{A}$ queries only in the variables $x_A, x_B, \alpha_A, \alpha_B, \beta_A, \beta_B, s, s_{(it)}, r, r_{(i,t)}$. Since the adversary's two different formal rational functions $\gamma/\eta, \gamma'/\eta'$ will have an accidental collision caused by the random choice of variables, the condition is set that no collision of this kind occurs in any of the groups G_0, G_T. A collision occurs if and only if for different rational functions γ/η and γ'/η' of the query satisfy $\gamma\eta' - \eta\gamma' = 0$. Note that the total degree of $\gamma\eta' - \eta\gamma'$ is in our case at most 5, the likelihood of this event, according to the Schwartz-Zippel Lemma [13,14], is $O(1/p)$. By a union bound, the probability of a collision occurring is maximized to $O\left(q^2/p\right)$ so that it can be calculated that there is a $1 - O\left(q^2/p\right)$ probability that no accidental collision will occur.

Conditional on the absence of collisions, if the challenger sets $\vartheta = \alpha \cdot s$, we must show that the opponents' views are identically distributed. In particular, under the above setup conditions, the only way to make an adversary's viewpoint different is two queries π, π' in G_0 such that $\pi \neq \pi'$ but $\pi|_{\vartheta=\alpha\cdot s} \neq \pi'|_{\vartheta=\alpha\cdot s}$. However, that case will not occur since only $e(g,g)^{\vartheta}$ is computed from the element ϑ in G_0. Thus, for some constant $\psi \neq 0$, one must have $\pi - \pi' = \psi\alpha s - \psi\vartheta$. When we provide a query a to the adversary, the adversary $\mathcal{A}$ will never be able to construct a query for $e(g,g)^{\psi\alpha s}$ as long as there is no accidental collision, which contradicts the above theory, thus proving the theorem.

4.3 Correctness Analysis

This subsection demonstrates the correctness of the scheme. First, this subsection shows the process of DO A decrypting the original ciphertext C_A using the original key Sk_A. Secondly, a correctness proof of the evolution of the ciphertext from the original ciphertext C_A to C_B is given. Finally, evidence of the correctness of the evolved ciphertext C_B in the decryption process is presented.

(1) Decryption of original ciphertext C_A: The original encrypted message and the public-private key pair obtained by the DO are $C_A = \left\{c, c', \forall_{\lambda_i \in Y}\left\{c_{(ij)}, \tilde{c}_{(ij)}\right\}_{j\in[0,\omega_i]}\right\}$, $Sk_A = \{x_A, \alpha_A, \beta_A\}$ and $Pk_A = \{y_A, h_A, f_A\}$ respectively. The decryption of original ciphertext is given as follows:

$$\frac{m \cdot {f_A}^s}{e\left(g, {h_A}^s\right)^{\alpha_A \cdot {\beta_A}^{-1}}} = \frac{m \cdot e(g,g)^{\alpha_A}}{e\left(g, g^{\beta_A}\right)^{\alpha_A \cdot {\beta_A}^{-1}}} = m$$

(2) The agent acquires the evolved ciphertext C_B: Upload the original ciphertext $C_A = \left\{c, c', \forall_{\lambda_i \in Y}\left\{c_{(ij)}, \tilde{c}_{(ij)}\right\}_{j\in[0,\omega_i]}\right\}$ and the re-encryption key $Sk_{A\to B} = \left\{k, k_{((i0))}, \tilde{k}_{((i0))}, \forall\, w \in P\left\{k_{(ij)}, \tilde{k}_{(ij)}, \tilde{\tilde{k}}_{(ij)}\right\}_{j\in(0,\omega_i']}\right\}$ to the cloud server. Only the process of calculating $\tilde{c}$ is given below, and c is the same as the original ciphertext:

$$\begin{aligned}
\tilde{c} &= \frac{e\left(\left(y_B{}^{\alpha_A} g^r\right)^{-\beta_A}, {h_A}^s\right)}{e(g,g)^{rs_{(ij-1)}} \cdot e\left(c_{(ij-1)}, k_{(ij)}\right) \cdot e\left(\tilde{c}_{(ij)}, \tilde{k}_{(ij)}\right)^{-1} \cdot e\left(c_{(ij)}, \tilde{\tilde{k}}_{(ij)}\right)} \\
&= \frac{e\left(\left(y_B{}^{\alpha_A} g^r\right)^{-\beta_A}, {h_A}^s\right)}{e(g,g)^{rs_{(ij-1)}} \cdot e\left(g^{s_{(ij-1)}}, g^{-r} H(w_i)^{r_{(ij)}}\right) \cdot e\left(\left(h_{(ij)}\right)^{s_{(ij)}} \cdot H(att(\lambda_i))^{s_{(ij-1)}}, g^{r_{(ij)}}\right)^{-1} \cdot e\left(g^{s_{(ij)}}, g^r \left(h_{(ij)}\right)^{r_{(ij)}}\right)} \\
&= \frac{e\left(\left(y_B{}^{\alpha_A} g^r\right)^{-\beta_A}, {h_A}^s\right)}{e(g,g)^{rs_{(ij-1)}} \cdot e\left(g^{s_{(ij-1)}}, g^{-r}\right) \cdot e\left(g^{s_{(ij-1)}}, H(w_i)^{r_{(ij)}}\right) \cdot e\left(\left(h_{(ij)}\right)^{s_{(ij)}}, g^{r_{(ij)}}\right)^{-1} \cdot e\left(H(att(\lambda_i))^{s_{(ij-1)}}, g^{r_{(ij)}}\right)^{-1} \cdot e\left(g^{s_{(ij)}}, g^r\right) \cdot e\left(g^{s_{(ij)}}, \left(h_{(ij)}\right)^{r_{(ij)}}\right)} \\
&= \frac{e\left(\left(y_B{}^{\alpha_A} g^r\right)^{-\beta_A}, {h_A}^s\right)}{e\left(g^{s_{(ij)}}, g^r\right)} = \frac{e\left(\left(g^{x_B \alpha_A} g^r\right)^{-\beta_A}, g^{\beta_A \cdot s}\right)}{e(g,g)^{r\cdot s}} = {f_A}^{s\cdot x_B}
\end{aligned}$$

(3) Decryption of delegated ciphertext C_B:When the user wants to access the ciphertext C_A, the agent will use the re-encryption key $Sk_{A\to B}$ of user B to obtain the evolved ciphertext C_B. Subsequently, the user decrypts it using his private key $Sk_B = \{x_B, \alpha_B, \beta_B\}$ and the $C_B = \left\{m \cdot {f_A}^s, f_A^{s \cdot x_B}\right\}$ obtained from the agent. The derivation process is as follows:

$$\frac{m \cdot {f_A}^s}{\left({f_A}^{s \cdot x_B}\right)^{-x_B}} = m$$

5 Performance Analysis

5.1 Performance Comparison

As shown in Table 2, it demonstrates the functionality and performance comparison of the proposed scheme with Xiong's scheme [16], Yao's scheme [6,15,17]. Let t_w,t_v,t_m,t_e and t_p denote the computation time of signature verification, ciphertext verification, multiplication operation, modular exponentiation, and pairing, respectively.

Table 2: Compared with other existing schemes

Scheme	ReEncryption ciphertext size	Decryption	Weighted	Immediate revocation
Scheme [15]	$8\|G_1\| + 1\|G_T\|$	$6t_p + 6t_e + t_w$	×	×
Scheme [16]	$5\|G_1\| + 2\|G_T\|$	$3t_p + t_e + t_v$	×	×
Scheme [17]	$2\|G_T\|$	$2t_p + t_m$	×	×
Scheme [6]	$7\|G_T\|$	$t_p + t_e + t_v$	×	√
Our scheme	$2\|G_T\|$	$t_m + t_e$	√	√

Firstly, Table 2 shows the comparison of two key features, namely "whether to support weighting of attributes" and "whether to support direct revocation by the user." In real-application scenarios, CSP usually requires fine-grained data sharing and efficient user revocation. Therefore, the two functions in Table 2 are important in cloud storage services today, and only the proposed scheme supports both functions. Second, the proposed scheme has high efficiency on the user side because the complex pairing operation in the previous decryption algorithms is replaced by the re-encryption algorithm, which is implemented by the CSP. The length of the ciphertext obtained at the user side is only 2 $|G_T|$. As shown in Table 2, decryption requires only power and multiplication operations, making the decryption cost much lower than other related schemes.

In summary, compared with other relative schemes, the proposed scheme has the smallest re-encrypted ciphertext size, which effectively enhances the communication efficiency on the user side, and adds additional functions such as weighting of attributes and attribute revocation so that the proposed scheme has a far wider range in the realm of health care.

5.2 Experimental Results

In this paper, the experimental environment is IntelliJ IDEA Community Edition 2023, and Java Pairing Based Cryptography Library and elliptic curves chosen as class A curves $y^2 = x^3 + x$ of F_q over the domain. It runs on a computer with parameters of 2.30 GHz Inter(R) Core (TM) i7-12700H, 16.0 GB of RAM, and operating system Windows 11.

The time consumed in running the decryption algorithm 100 times in the simulation experiment is listed as shown in Fig. 3a. The maximum time required to decrypt the ciphertext is 8.9 ms, and the minimum is 6.03 ms. The time consumed for user decryption is kept in the range of 6–9 ms and does not grow with the magnitude of user attributes. In addition, we perform communication overhead tests of the scheme's decryption algorithm on a smartphone with a 2.4 GHz CPU and 8 GB RAM. The experimental results demonstrate that the communication overhead for the user is 2 $|G_T|$, which is about 0.8 kb in the implementation.

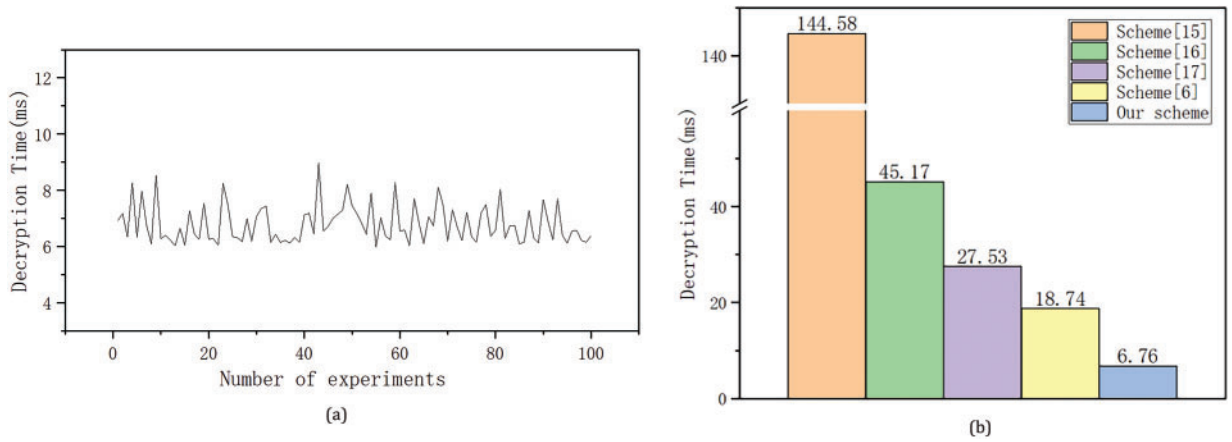

Figure 3: (a) Decryption time cost; (b) Comparison of decryption time cost [6,15–17]

We use the average of one hundred decryption times as the decryption time for this scheme. As can be seen in Fig. 3b, the proposed scheme has the least user decryption time overhead compared to other schemes. Although this scheme is constructed based on bilinear mapping, the decryption algorithm does not have a pairing operation with high overhead. Therefore, the proposed scheme has less computation and communication overhead on the user side, making it feasible for users to access cloud servers with resource-constrained devices.

6 Conclusion

In order to make cloud servers satisfy as high as possible access control mechanisms, a WAB-CPRE scheme is proposed. By combining weighted attribute-based encryption, the proposed scheme provides more granularity and a more flexible access structure for decryption. The scheme also supports CSP in performing direct revocation of user access rights, and there is no computational cost to achieve revocation. In addition, the user side of the scheme has low computational and communication overhead, enabling users to access cloud service data even in device-constrained environments. In the future, our motivation is to construct a more effective CPRE scheme where the DO can directly revoke a user's weighted attributes.

Acknowledgement: Thanks to my tutor for careful guidance and the help of my classmates.

Funding Statement: Programs for Science and Technology Development of Henan Province, grant number 242102210152. The Fundamental Research Funds for the Universities of Henan Province, grant number NSFRF240620. Key Scientific Research Project of Henan Higher Education Institutions, grant number 24A520015. Henan Key Laboratory of Network Cryptography Technology, grant number LNCT2022-A11.

Author Contributions: Xixi Yan: the main idea, conceptualization and methodology. Jing Zhang: The framework was built, the encryption scheme was written, the security proof was derived, and the experimental analysis and test were carried out. Pengyu Cheng: Integration and analysis of experimental data. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The authors confirm that the data supporting the findings of this study are available within the article.

Ethics Approval: Not applicable.

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

References

1. Weng J, Deng RH, Ding X, Chu CK, Lai J. Conditional proxy re-encryption secure against chosen-ciphertext attack. In: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security; 2009; Sydney Australia: ACM. p. 322–32. doi:10.1145/1533057.1533100.
2. Blaze M, Bleumer G, Strauss M. Divertible protocols and atomic proxy cryptography. In: Advances in cryptology-EUROCRYPT'98. Berlin, Heidelberg; 1998. p. 127–44.
3. Elgamal T. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans Inf Theory. 1985;31(4):469–72. doi:10.1109/TIT.1985.1057074.
4. Zhao J, Feng D, Zhang Z. Attribute-based conditional proxy re-encryption with chosen-ciphertext security. In: 2010 IEEE Global Telecommunications Conference GLOBECOM 2010; 2010 Dec 6–10; Miami, FL, USA: IEEE; 2010. p. 1–6. doi:10.1109/GLOCOM.2010.5684045.
5. Yang Y, Zhu H, Lu H, Weng J, Zhang Y, Choo KR. Cloud based data sharing with fine-grained proxy re-encryption. Pervasive Mob Comput. 2016;28(1):122–34. doi:10.1016/j.pmcj.2015.06.017.
6. Yao S, Dayot RVJ, Kim HJ, Ra IH. A novel revocable and identity-based conditional proxy re-encryption scheme with ciphertext evolution for secure cloud data sharing. IEEE Access. 2021;9:42801–16. doi:10.1109/ACCESS.2021.3064863.
7. Yang H, Li L, Yang C. A fine-grained certificateless conditional proxy broadcast re-encryption scheme without pairing. In: 2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC); 2022 Jun 17–19; Chongqing, China: IEEE; 2022. p. 1414–23. doi:10.1109/ITAIC54216.2022.9836814
8. Tang Y, Jin M, Meng H, Yang L, Zheng C. Attribute-based verifiable conditional proxy re-encryption scheme. Entropy. 2023;25(5):822. doi:10.3390/e25050822.
9. Wang Y, Wang M. Improved AB-CPREs with revocability and HRA security under LWE. IET Inf Secur. 2024;2024(1):4333883. doi:10.1049/2024/4333883.
10. Liu X, Ma J, Xiong J, Li Q, Ma J. Ciphertext-policy weighted attribute based encryption for fine-grained access control. In: 2013 5th International Conference on Intelligent Networking and Collaborative Systems; 2013 Sep 9–11; Xi'an, China: IEEE; 2013. p. 51–7. doi:10.1109/INCoS.2013.18.
11. Waters B. Functional encryption for regular languages. In: Annual International Cryptology Conference; 2012; Berlin/Heidelberg: Springer; p. 218–35.
12. Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy (SP '07); 2007 May 20–23; Berkeley, CA, USA: IEEE; 2007. p. 321–34. doi:10.1109/SP.2007.11.
13. Schwartz JT. Fast probabilistic algorithms for verification of polynomial identities. J ACM. 1980;27(4):701–17. doi:10.1145/322217.322225.
14. Zippel R. Probabilistic algorithms for sparse polynomials. In: Proceedings of the International Symposiumon on Symbolic and Algebraic Computation; 1979; Berlin/Heidelberg: Springer. p. 216–26.
15. Yao S, Dayot RVJ, Ra IH, Xu L, Mei Z, Shi J. An identity-based proxy re-encryption scheme with single-hop conditional delegation and multi-hop ciphertext evolution for secure cloud data sharing. IEEE Trans Inf Forensics Secur. 2023;18(2):3833–48. doi:10.1109/TIFS.2023.3282577.

16. Xiong H, Wang Y, Li W, Chen CM. Flexible, efficient, and secure access delegation in cloud computing. ACM Trans Manage Inf Syst. 2019;10(1):1–20. doi:10.1145/3318212.
17. Yao S, Sankar R, Ra IH. A collusion-resistant identity-based proxy reencryption scheme with ciphertext evolution for secure cloud sharing. Secur Commun Netw. 2020;2020:8833693. doi:10.1155/2020/8833693.