IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 28, 2019, accepted November 6, 2019, date of publication November 11, 2019,
date of current version November 27, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2952640

Progressive Image Restoration Based on CP-ABE With Constant-Size Ciphertext and Constant Bilinear Calculation

HUAIBO SUN, HONG LUO, AND YAN SUN
Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Hong Luo (luoh@bupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772085, Grant 61532012, Grant 61672109, and Grant 61877005.

ABSTRACT Sharing photos on an open social platform with one-to-many features increases the leakage risk of information of the shared photos. Ciphertext-policy attribute-based encryption (CP-ABE) is a one-to-many public key cryptosystem, and its access policy can be set by data owners. Therefore, if we share the protected image on the social platform and encrypt the restoration parameters of the shared image with CP-ABE, then only the accessor whose attribute set satisfies the access policy of CP-ABE can decrypt the ciphertext corresponding to his/her attribute level and then restore the image with the corresponding sharpness level. Consequently, image information protection is realized in a one-to-many scenario. However, for the existing CP-ABE, only when further shortening its ciphertext, reducing the number of bilinear pairing calculations during decryption, and ensuring flexible access policies is it beneficial for CP-ABE to be widely used in popular smart terminals that can run social software. Furthermore, existing algorithms for restoring images also have room for improvement in terms of time and space requirements. For this reason, we propose a progressive image restoration strategy based on the improved CP-ABE in this paper. For the improvement of CP-ABE, we design a new encryption scheme to achieve a shorter constant-size ciphertext, construct an auxiliary function to help the independent authority centers generate private keys for users in new ways, design a decryption algorithm with only one bilinear pairing calculation, and provide the update algorithms for attribute revocation. Subsequently, we build a distributed CP-ABE based on our improved CP-ABE, and the correctness and security of the proposed CP-ABE algorithm are also proven. In addition, we follow the tree access policy to support the access policy, including AND, OR, NOT and threshold operations simultaneously, to adapt to the requirement of some user attributes that undergo frequent changes. Moreover, based on the atmospheric scattering model, we adopt three algorithms to restore the protected image in the dehazing mode according to the sharpness parameters. Compared with the existing literature, we provide users with not only a flexible and efficient access control strategy, shorter constant-size ciphertext and fewer bilinear calculations but also multiple algorithms for restoring images. The experimental results for six privilege levels demonstrate the superiority of our algorithms in protecting image information.

INDEX TERMS CP-ABE, constant-size ciphertext, bilinear pairing calculation, progressive image restoration, atmospheric scattering model.

I. INTRODUCTION

Currently, more social software can run on smart devices, and an increasing number of functions are endowed to various social software, such as Twitter, QQ, and WeChat. This ability makes it possible for people to share images, videos, and even very private items on social platforms at any time using smart devices, even when at work, which obviously increases the risk of information leakage because of the openness of social platforms. However, if the data owner can control the set of accessors and their privileges independently [2]–[5] and restore the image information (progressively clear) by gradation according to the accessors' level of privilege [6]–[14], it will more effectively protect the security of the image or

The associate editor coordinating the review of this manuscript and approving it for publication was Naveed Akhtar.

video shared on a social platform. CP-ABE is a one-to-many public key cryptosystem, and its access policy can be set by the data owner to achieve fine-grained access control. Therefore, if the picture is first protected and then shared on the social platform, and then the sharpness parameters of the picture are encrypted by CP-ABE, then only the accessor who has the attribute set satisfying the access policy of CP-ABE can decrypt the ciphertext of sharpness parameters corresponding to his/her privilege level. Then, the algorithm can be called for restoring the image to the corresponding sharpness; thus, the security of shared image information can be more reasonably guaranteed based on CP-ABE.

However, due to the large-scale bilinear pairing calculations and long ciphertext for the CP-ABE proposed in [1], the application of CP-ABE on smart devices is limited, even though the tree access policy proposed in [1] can support a flexible access structure. Therefore, even though CP-ABE is used in [2], [3], the performance of the algorithms they proposed is restricted due to the high complexity of CP-ABE in terms of time and space. In addition, the performance of existing algorithms for restoring images also needs to be improved in terms of time and space.

Many researchers have proposed CP-ABEs with novel access strategies and (or) constant-size ciphertexts to expand their application scopes, for example, the CP-ABE supporting linear secret-sharing schemes (LSSS) [15]–[18], the CP-ABE supporting conjunctive normal form (CNF) [19], the CP-ABE supporting the threshold policy [20], [21], and the CP-ABE supporting the AND-gate policy [22]–[28].

However, to the best of our knowledge, in the existing literature, the access control strategies can still be further perfected because the algorithms with the shorter ciphertext and the smaller number of bilinear pairing calculations either support AND, OR, NOT, and the wildcard policy but not the threshold policy [19]; support only the threshold policy [20], [21]; or support only the AND-gate policy [22]–[28]. In addition, in the practical application of an access control strategy, there are scenarios in which AND, OR, NOT, and the threshold operation appear simultaneously. For example, as shown in Fig. 1, these operations are included in the attribute policy that must be satisfied by people who want to access the photo, where the node "Schoolmate" extends the definition of classmate-friend (that is, university friends but not classmates, or desk mates but not in university) through its three subnodes, so that the meaning of classmate-friend better matches real life relationships and "¬ Friends of B Group" can be more targeted to remove the access privileges of some people from classmate-friends (here, ¬ means non). Moreover, the algorithms that can support AND, OR, NOT, and threshold operations simultaneously can further shorten their ciphertext lengths (whether using a tree access structure [1], [29], [30] or a policy matrix ($\mathcal{M}$, p) [15]–[18]).

Therefore, to further expand the application scope of CP-ABE, especially the use in intelligent terminals, a lightweight CP-ABE is needed to support AND, OR, NOT, and the threshold policies (called the perfect access control strategy in this paper), and it should have shorter ciphertext and fewer bilinear pairing calculations during decryption.

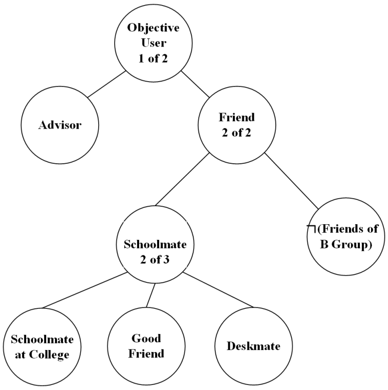


FIGURE 1. Attribute policy that should be satisfied by the attributes of legal photo accessors.

To this end, in this paper, an improved CP-ABE with constant-size ciphertext and a constant number of bilinear pairing calculations is first proposed based on the CP-ABE in [1]. Then, a distributed CP-ABE is built based on our improvement, and the update algorithms for attribute revocation are also provided. Then, we prove the correctness and security of the proposed CP-ABE algorithm. Next, after providing three algorithms for restoring the image, we design a progressively clear image restoration algorithm based on our distributed CP-ABE and the algorithms for restoring the image to protect the security of image information shared on a social platform. Moreover, this algorithm takes the image sharpness parameter as the plaintext of our CP-ABE. Therefore, when the accessors request the image, this algorithm first decrypts the corresponding image sharpness parameters based on the accessor's attribute set, and then restores the images with the corresponding sharpness using an algorithm for restoring the image. For the improvement of CP-ABE, we design a new encryption scheme that can generate a shorter ciphertext, and propose an auxiliary function to help the authority centers generate private keys for image accessors. Additionally, we design a new method to generate the private keys, and in response to the above algorithms, design a new scheme without the bilinear pairing calculations to determine the privilege level of the image accessor when we decrypt according to the tree access policy. Consequently, there is only one bilinear pairing calculation during decryption. Moreover, we design the update algorithms for the attribute revocation, build a distributed CP-ABE based on our improved CP-ABE, and prove the correctness and security of the proposed CP-ABE algorithm. To progressively restore an image, considering the shortcomings of existing algorithms in terms of restoration accuracy and storage space requirements, three algorithms are provided to perform loss or lossless image restoration in dehazing mode based on the atmospheric scattering model.

The algorithms proposed by us better adapt our image sharing to the current context in which people interact on social platform anytime and anywhere using smart devices because the attributes of image accessors may change at any time in this case. More than 6000 experiments demonstrate that our algorithms achieve the correct access control (that is, restore the corresponding sharpness parameters of images for users) and have a shorter constant-size ciphertext, constant bilinear pairing calculation during decryption, and good image restoration effect. Compared to the existing algorithms, our algorithms have better performance.

The main contributions of this article are as follows:

1) A distributed CP-ABE is built based on our improved CP-ABE, which has a shorter constant-size ciphertext and only one bilinear pairing calculation when decrypting, and the update algorithms are provided to realize attribute revocation under the distributed CP-ABE scheme.

2) The correctness and security of the proposed CP-ABE algorithm are proven.

3) Three algorithms are provided for progressively and clearly restoring images.

4) A progressively clear image restoration algorithm is designed based on our distributed CP-ABE scheme and the algorithms for restoring images, in which the distributed CP-ABE is applied to progressive image restoration for the first time.

The rest of this paper is organized as follows. Related works are discussed in Section II. Background knowledge is provided in Section III. The system model, system framework, and security model of CP-ABE proposed in this paper are introduced in Section IV. Our proposed CP-ABE is discussed in Section V. The correctness and security of our CP-ABE method are demonstrated in Section VI. We design a progressively clear restoration scheme for an image based on our CP-ABE scheme and the algorithms for progressively and clearly restoring the images in Section VII. The experiments implemented and discussed are in Section VIII. Finally, Section IX finishes this article with the conclusion and suggestions for future work.

II. RELATED WORK

To ensure that the confidentiality of the data is not compromised, Bethencourt *et al.* [1] proposed the CP-ABE to realize complex access control on encrypted data, and their methods were secure against collusion attacks. Since CP-ABE was proposed, this algorithm has been continuously improved and applied in image information restoration. For this reason, next, we provide the related work in two areas: image restoration and CP-ABE.

A. IMAGE RESTORATION

To solve the problem of information security for shared images on social platforms, Yuan *et al.* [2] proposed using traditional public key cryptography and CP-ABE to realize privacy-preserving JPEG image sharing, in which only those people accessing the image whose attribute sets satisfy the CP-ABE access policy can view the original appearance of the disturbed region of interest (ROI) in the image. Considering the use of Diaspora as a distributed social network, Picazo-Sanchez *et al.* [3] provided a multi-authority attribute-based encryption (MA-ABE) and allowed access control by attaching policies to photos based on their MA-ABE. Sun *et al.* [5] proposed hiding the information in the ROI out of the ROI area and then generating an image mosaic in the ROI area. When an accessor is accessing the image information, with the help of CP-ABE, the system first calculates the privilege level of the accessor according to his or her attribute set and then adaptively restores the corresponding mosaic blocks according to the privilege level, thus reasonably restoring the secret information of the image. In addition, the author also set up a vote attribute (VA) to achieve rapid user revocation. It can be seen that CP-ABE is used to restore images in all the literature mentioned above, but it has not achieved progressive image restoration.

For progressive image restoration, Zhai *et al.* [6] proposed a unified framework to implement progressive image restoration based on mixed Laplace regular regression. Wang *et al.* [7] designed a generative adversarial network to improve the quality of restored images synchronously with the progressive multi-level design principle. Wang *et al.* [8] designed a Laplacian-pyramid-based convolutional network architecture to predict, under different resolutions, the data of the missing region in a face image, and constructed a new residual learning network that progressively removed the color difference between the missing area and the surrounding area to gradually improve the facial image. Some current algorithms have good performance only when restoring fixed corruption-level images; to address this, Gao and Grauman [9] proposed an on-demand learning algorithm based on a convolutional neural network to train the image restoration model, thus self-generating the training instances for different difficulty levels using a feedback mechanism. Knaus and Zwicker [10] used deterministic annealing techniques to progressively denoise the image through a simple physical process.

The loss image restoration techniques provided by the abovementioned literature all have good performance. However, some researchers have studied progressive image restoration techniques without distortion. For example, Yan *et al.* [11] derived three progressive secret image sharing (PSIS) algorithms from the traditional secret image sharing (SIS) scheme using the threshold mechanism: (k, n) PSIS, $(k1, n)$ PSIS, and $(k1, k2, n)$ PSIS. These algorithms first divided the secret image into k, $k1$, or $k1, k2$ blocks and then gradually restored them according to the corresponding threshold policy. These algorithms do not introduce pixel expansion as the traditional SIS algorithms do, and their main concern is global progressiveness. Chao and Fan [12] proposed a random-grid-based progressive visualization secret-sharing scheme; in this scheme, each user participating in the secret sharing could adjust the priority of the shared image block according to their security decision level. During restoration, the secret image could be restored to different extents based on the priority of the stacked shares. At the same time, the priority level of each shared image block could not be distinguished according to the average light transmission of the reconstructed image block, thereby ensuring the security of the image information. Liu *et al.* [13] proposed an SIS mechanism in which the threshold was alterable, and each participant needed only to keep one initial shadow. When reconstructing an image, the dealer determined the threshold (k, n) based on the security level. If the threshold was unchanged, then the image could be restored with greater than or equal to k initial shadows; if the threshold was increased or decreased, the dealer needed to post additional information, and each participant needed to update their shadows accordingly, so that the updated threshold of the shadows could be changed accordingly. Their solution met the demand for a dynamic security environment and could selectively restore the shadows of an image. It can be seen that although the specific methods are different, the schemes used in the above studies all belong to the SIS scheme of the (k, n) threshold type, and the original image can be recovered when the number of shadow images reaches or exceeds k. However, this method requires multiple shadow images as the raw material to restore different degrees of image sharpness for users, which obviously increases the space requirements of the algorithm.

Of course, the algorithm provided in [14] did not require the shadow images. The receiver of their algorithm extracted the bits of three least significant bit (LSB)-layers from the pixel sets divided into the square set, the triangle set and the circle set, using the embedded secret key for progressive image restoration. The algorithm could restore the original image after three rounds of extraction. However, when performing their algorithm, the secret information needed to be extracted progressively from the encrypted carrier image, which would cause decryption loss for the encrypted image due to the inappropriate parameters, and the maximal embedding rate of this algorithm was only 0.033 bpp.

B. CP-ABE

Cheng *et al.* [15] provided a CP-ABE that supported improved access strategies and demonstrated that their construction method satisfied selective security under the decisional bilinear Diffie-Hellman assumption. Yang *et al.* [17] implemented a CP-ABE that could be efficiently updated by an outsourcing strategy that updates to the server. They also designed effective policy update algorithms for the access structures of Boolean formulas, LSSS structure, and access tree types, which enabled more efficient access control for big data stored in the cloud. Dong *et al.* [18] used an LSSS access structure $(\mathcal{M}, p)$ to provide an efficient, scalable and privacy-preserving semantic security strategy by using CP-ABE in combination with identity-based encryption (IBE) technology. In addition to protecting the security of large-scale shared data, their policy could protect the privacy of cloud users and support efficient and secure dynamic operations such as user revocation and attribute modification. The algorithms provided in the above literature can support the perfect access control strategy, but their ciphertexts are not of constant size.

There are some algorithms that provide constant-size ciphertext. For example, Canard and Trinh [19] proposed a private CP-ABE that supported the CNF access policy, which had only one restriction: the single attribute could appear at most k_{max} times in the access formula. The authors of [19] provided two schemes, namely, a scheme that satisfied the basic selective security in the standard model and a scheme that satisfied the chosen ciphertext attack security in the random oracle model; the lengths of the ciphertexts under both schemes were all of constant size. However, we find that such a policy is similar to the AND-gate policy. Nevertheless, in real life, the length of a CNF formula is uncertain, and formulas with the same length do not always correspond to the same access privilege.

Moreover, considering that ciphertext size depended linearly on the number of attributes contained in the access policy, Herranz *et al.* [20] proposed a threshold-based ABE that required accessors to have at least t attributes of the universe attribute set if the accessors wanted to successfully decrypt the ciphertext, and the ciphertext was of constant size. To address the situation in which the entire system may crash after an attack on the trusted authorization center of the centralized MA-ABE, Lin *et al.* [21] proposed a threshold multi-authority fuzzy identity-based encryption scheme. Their scheme did not have a central authority, but it required at least $t + 1(t < \frac{n}{2})$ authority centers to be honest, and the user must have at least d_k attributes of the given attribute set to decrypt the ciphertext. The ciphertext they provided has a constant size. Zhang *et al.* [24] proposed hierarchical MA-ABE on prime order groups. They constructed an AND-gate access structure, used wildcards to represent some ‘‘don't care'' attributes of the user and provided constant-size ciphertext. Li *et al.* [26] proposed an attribute-based access control scheme with two-factor protection for data sharing in a multi-authority cloud storage system. The scheme they proposed adopted an AND-gate access structure, provided constant-size ciphertext and incurred low computation cost. Odelu *et al.* [27] designed an RSA-based CP-ABE scheme with AND-gate access structures that had a constant-size secret key and ciphertext. The time complexity of decryption and encryption was $O(1)$, and RSA was used to generate the large integers used in their CP-ABE. There were no bilinear pairing calculations in their algorithm. In addition, Xiong *et al.* [28] used an AND-gate access structure with wildcards to construct their CP-ABE access policy, but the ciphertext output from their algorithm was not constant in size.

In addition to this, Ostrovsky *et al.* [29] constructed an ABE scheme with non-monotonic access structures that allowed a user's private key to be expressed according to any access formula over attributes. To improve the performance and efficiency of CP-ABE, Li *et al.* [30] changed the access

TABLE 1. The symbols used in this article and their meanings.

Symbols	Meanings
$S^v(S)$	A set of attributes that requires the value of the attribute to be distinguished when it is being used.
$\xi(i,j)$	The light propagation rate in the atmospheric scattering model
GP	The global public key of the CP-ABE system
PK_{AID}	The public key issued by AC AID
SK_{AID}	The secret key kept by AC AID
U_{AID}	The set of all the attributes governed by AC AID
S_{AID}	The attribute set of some data accessor governed by AC AID
SK_{user}	The secret key of some data accessor

* For ease of understanding, next, we use S to represent the set of attributes in this article, and no longer emphasize the v in S^v. However, when using the concrete attribute, we still distinguish the values of different attributes.

structure in CP-ABE, and presented a new CP-ABE system based on the ordered binary decision diagram. However, their ciphertexts are not of constant-size.

III. PRELIMINARIES

In this section, we provide the basic knowledge used in this article.

A. SYMBOLS AND THEIR MEANINGS

In this section, we provide the symbols used in this article and their meanings, as shown in Table 1, where *AID* refers to the identity of the authority center (AC).

B. BILINEAR MAPS

Let G_1 and G_2 be the multiplicative cyclic groups whose orders are prime p, g is the generator of G_1, and e is a bilinear map: $G_1 \times G_1 \rightarrow G_2$. e has the following characteristics:

1. Bilinearity: for all $u, v \in G_1$ and $a, b \in Z_p$, there is $e(u^a, v^b) = e(u, v)^{ab}$.

2. Non-degeneracy: $e(g, g) \neq 1$.

3. Computability: the group operation and bilinear map $e : G_1 \times G_1 \rightarrow G_2$ on G_1 can be effectively calculated, and then G_1 is called a bilinear group.

4. e has the following symmetry: $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

C. ACCESS STRUCTURE

Definition 1 (Monotone Access Structure [1]): Let $\{P_1, P_2, \ldots, P_n\}$ be a set of parties. For $\forall \boldsymbol{B}, \boldsymbol{C}$, if $\exists \boldsymbol{B} \in \boldsymbol{A}$ and $\boldsymbol{B} \subseteq \boldsymbol{C}$, then $\boldsymbol{C} \in \boldsymbol{A}$, and then the collection $\boldsymbol{A} \subseteq 2^{(P_1, P_2, \ldots, P_n)}$ is monotonic.

An access structure (especially a monotone access structure) is a collection (monotone collection) $\boldsymbol{A}$ of the non-empty subsets of $\{P_1, P_2, \ldots, P_n\}$, such as $\boldsymbol{A} \subseteq 2^{\{P_1, P_2, \ldots, P_n\}} \setminus \{\varnothing\}$. The sets in collection $\boldsymbol{A}$ are called the authorization sets, and the sets not in $\boldsymbol{A}$ are called the unauthorized sets.

Definition 2 (LSSS Structure [29], [31]): A secret-sharing scheme $\boldsymbol{\Psi}$ on the collection of parties is called linear(over Z_P) if

1. The sharing provided for each party forms a vector on Z_P.

2. There is a share-generating matrix $\mathcal{M}$ for $\boldsymbol{\Psi}$, and it has l rows and $n+1$ columns. For all $i = 1, 2, \ldots, l$, the i^{th} row of $\mathcal{M}$ is marked by one of $\rho(i)$, where ρ is a mapping function from $\{1, 2, \ldots, l\}$ to Z_p. When we consider the column vector $\vec{v} = (s, r_1, r_2, \ldots, r_n)$, where $s \in Z_p$ is the secret to be shared and $r_1, r_2, \ldots, r_n \in Z_p$ are randomly selected, then $\mathcal{M}\vec{v}$ is the vector of l shares of s according to $\boldsymbol{\Psi}$.

Definition 3 (Access Tree Structure [1], [17]): An access tree is defined as tree $\mathcal{T}$, where each non-leaf node x represents a trapdoor (k_x, num_x) (num_x is the number of children of the non-leaf node), and each leaf node x is described by an attribute $att(x)$ and a threshold k_x. The function $parent(x)$ represents the parent node of x in the access tree. $\mathcal{T}$ also defines a sequence number for each child node, and the function $index(x)$ returns the sequence number of node x.

In our construction, the internal nodes of the access tree can be AND-gate, OR-gate, NOT-gate, or threshold operations. In this article, we do not consider the access policy tree that supports wildcards (also called "don't care"); we believe that since the wildcard policy is also called "don't care", even if the access policy tree contains a set of attributes expressed by a wildcard policy, there is no need to care about the impact of these attributes on the user's privilege.

Satisfying an Access Tree: Let $\mathcal{T}$ denote an access tree whose root is R_n and $\mathcal{T}_x$ represent the subtree rooted at node x in $\mathcal{T}$; therefore, $\mathcal{T}$ can be expressed as $\mathcal{T}_{R_n}$. If the attribute set S satisfies the access tree $\mathcal{T}_x$, we denote it by $\mathcal{T}_x(S) = 1$. We calculate $\mathcal{T}_x(S)$ recursively as follows:

If x is a non-leaf node, then the value of $\mathcal{T}_{x'}(S)$ of the child node x' of node x is evaluated according to the operational characteristics of x; if x is a leaf node, then $\mathcal{T}_x(S) = 1$ if and only if $att(x) \in S$.

In this paper, when establishing the access control strategy, we allow image accessors to have negative attributes. For example, among the students who belong to some high school, only those whose attribute sets satisfy the conditions "(height > 175cm $\wedge$ age < 20 years old $\wedge$ graduating students) $\vee$ (height > 170cm $\wedge$ age < 18 years old $\wedge$ $\neg$graduating students) $\vee$ ($\neg$graduating students, often exercise, age >15, 2of 3)", are allowed to visit this sport photo exhibition to prepare for the next sporting event.

D. DECISIONAL Q-BILINEAR DIFFIE-HELLMAN EXPONENT (Q-BDHE) ASSUMPTION

G_1 and G_2 are p-order cyclic groups (p is a prime), g is a generator of G_1, and $e : G_1 \times G_1 \rightarrow G_2$ is a bilinear map. Randomly select two elements a and s from Z_p, and calculate

$y_{g,a,q} = \{g, g^s, g^a, g^{a^2}, \ldots, g^{a^q}, g^{a^{q+2}}, \ldots, g^{a^{2q}}\}$. If no algorithm can distinguish $e(g, g)^{a^{q+1}s}$ and a random element exists in G_2 with non-negligible probability in polynomial time, then the decisional q-BDHE assumption is true.

E. LAGRANGE COEFFICIENT

Given $n + 1$ points (x_i, y_i), we can uniquely determine an n-order polynomial f that is calculated as follows:

$$f(x) = \sum_{i=1}^{n+1}(y_i \cdot \prod_{1\leq j\leq n+1, j\neq i} \frac{x - x_j}{x_i - x_j}). \tag{1}$$

For $i \in Z_p$, $S \subseteq Z_p$, we define the Lagrange coefficient as

$$\mathop{\Delta}_{i,S}(x) = \prod_{j\in S, j\neq i} \frac{x - j}{i - j}.$$

F. ATMOSPHERIC SCATTERING MODEL

To facilitate an understanding of the algorithms for restoring an image, which is based on the atmospheric scattering model, we apply this model to a specific image [32], [33] as follows:

$$I(i, j) = J(i, j) * \xi(i, j) + \Delta_\infty * (1 - \xi(i, j)), \tag{2}$$

where (i, j) represents the spatial coordinate of an image pixel, $I(i, j)$ is the observed foggy image; $J(i, j)$ is the clear fog-free image; Δ_∞ is the atmospheric light intensity at infinity, which is usually a global constant and unrelated to spatial coordinates; and $\xi(i, j)$ denotes the propagation rate of light, and its value is in [0, 1]. In a homogeneous medium, the propagation rate of light can be expressed as

$$\xi(i, j) = e^{-\varphi * \zeta(i,j)}, \tag{3}$$

where $\zeta(i, j)$ is the depth of field of the scene, and φ is the scattering coefficient of the atmosphere. When hazing an image, hazing effects with different concentrations are realized by the different values of φ.

G. CP-ABE IN [1]

In this section, to facilitate an understanding of the content we have provided in CP-ABE, we introduce the CP-ABE proposed in [1] in detail.

CP-ABE uses a tree-type policy to specify the accessors who have privilege to access the private data, and the policy tree can contain AND, OR, NOT, and threshold operations, thus making it easy to develop various types of access policies.

There are four basic algorithms in CP-ABE, in which *Setup*() is used to generate the public key and the master key, and *Encrypt*() is used to encrypt the plaintext, but the ciphertext length is linear with the number of attributes in the policy tree. *KeyGen*() is used to generate a private key for an accessor. *Decrypt*() is used to decrypt the ciphertext for the accessor whose attribute set satisfies the access policy, but in the decryption process, when an attribute is a leaf node of the policy tree, the algorithm needs to execute two bilinear pairing operations. A fifth algorithm, *Delegate*(), can also be selected, which is used to re-randomize the key directly received from the authority. In addition, CP-ABE builds the system in a central architecture.

Analogous to KP-ABE, CP-ABE is also an extension algorithm for attribute-based encryption. However, in KP-ABE, the key issuer controls the users who can access the data, whereas in CP-ABE, the encryptor decides who can access the encrypted data. Therefore, CP-ABE is more in line with the actual requirements.

IV. THE SYSTEM MODEL, SYSTEM FRAMEWORK AND SECURITY MODEL OF CP-ABE

In this section, we present the system model, system framework, and security model for CP-ABE proposed in this paper.

A. SYSTEM MODEL

In this article, we consider a cloud storage system with multiple ACs. The system model includes the following entities: trusted third party (TTP), attribute ACs, servers, data owners (owners), and data accessors (users, that is, the image accessors as mentioned above)

1) TRUSTED THIRD PARTY

The TTP generates the global parameters, and based on the random numbers saved on it, performs policy tree-based user privilege determination; this determination is the foundation of the final decryption.

2) AUTHORITY CENTER

Each attribute authority center is independent and is responsible for the attribute management in its own domain. Simultaneously, they also generate the public/private key pairs for each attribute in their management domain and generate the private keys for each user based on his/her attributes.

3) SERVER

The server stores the data coming from the data owner and provides data access services to the user. It is also responsible for updating the ciphertext from that corresponding to the old access policy to that corresponding to the new access policy. Moreover, the server is also required to update the access policy corresponding to the encrypted data stored in the cloud.

4) OWNER

The data owner defines the access policy and encrypts the data under that access policy before depositing the data to the cloud. At the same time, the data owner generates an auxiliary function and asks each attribute AC to use this auxiliary function to generate the private key for the user. The data owner also sets the corresponding number of access policies according to the privilege levels he or she set and stores them on the server for a policy update including attribute revocation. Additionally, the data owner stores the random number used in the encryption process to the TTP for the calculation during the policy tree-based privilege determination before the final decryption.

5) USER

Each user (data accessor) can obtain the ciphertext from the server. When decrypting, the TTP determines privileges before the final decryption. We do not set a globally unique identity for the user in the server but instead determine whether the user has the privilege to decrypt the data based on his or her attributes, whose values are variable at any time. We assume in this paper that the attribute set does not have a one-to-one correspondence with the level of privilege held by the user. That is, the privilege level of the user may not be high, although his/her attribute set contains many elements. Of course, for the same user, more attributes will mean a higher access privilege. Since a certain attribute set must correspond to a specific privilege level, we use the attribute level and privilege level indiscriminately below.

B. SYSTEM FRAMEWORK

The algorithms included in the proposed system framework can be divided into two categories: basic algorithms and update algorithms when attributes are revoked, where *GSetup*(), *AuthSetup*(), *SEncrypt*(), *conSekey*(), *SKeyGen*(), and *SDecrypt*() are applied to the basic encryption and decryption; for example, *SEncrypt*() and *SDecrypt*() are used for encryption and decryption, *SKeyGen*() is used to generate a private key, and *conSekey*() is used as an auxiliary algorithm for *SKeyGen*(). *UPPolicy*(), *UPconSekey*(), *UPKeyGen*(), and *CTUpdate*() are applied to the update operation when the attributes are revoked. For example, *UPPolicy*() is used to update the old policy tree to the new policy tree, and feedback the update results to the algorithms that need these results; *UPKeyGen*() and *CTUpdate*() are used for private key update and ciphertext update after attribute revocation, respectively; and *UPconSekey*() is an update algorithm of *conSekey*(), which is used as an auxiliary algorithm for *UPKeyGen*(). Next, we give an introduction to them. Their implementation processes are provided in Section V.

1) $GSetup(1^\lambda) \rightarrow GP$

The global setup algorithm for the CP-ABE system. It takes as input the global safety parameter 1^λ, and sets the global public key GP.

2) $AuthSetup(GP, U_{AID}) \rightarrow (PK_{AID}, SK_{AID})$

The setup algorithm for the AC. Each of the ACs generates a public/private key pair (PK_{AID}, SK_{AID}) for itself based on the global public key GP and the attribute set U_{AID} governed by its own identity AID.

3) $SEncrypt(GP, \{PK_{AID}\}, M, \mathcal{T}) \rightarrow CT$

The encryption algorithm. It takes as input the global public key GP, the public key set $\{PK_{AID}\}$ belonging to some ACs, the message M (represented by a long integer or a string) representing the parameter of image sharpness level, and the access policy $\mathcal{T}$, and outputs the ciphertext $CT = \{\mathcal{T}, C, C'\}$. Then, the encryptor generates N random numbers $r_i \in Z_p$ to construct an auxiliary function $conSekey(GP, \{PK_{AID}\}, S_c, tag)$. Here, N represents the number of attributes contained in the policy tree; S_c denotes the set of attributes to be processed by the function *conSekey*(); *tag* is the tag of user's attributes, which is used to help record the number of attributes with the same *tag* that are processed within a given period of time when the private key is calculated for the users' attribute. When the time period expires, the function *conSekey*() sends the expression h_1' of the access policy (which also corresponds to the attribute level), to which the attribute set marked by the *tag* should belong, to the governor of the last attribute while sending the private keys.

4) $SKeyGen(GP, S_{AID}, SK_{AID}, S_{user}, AID) \rightarrow SK_{user}^{AID}$

The algorithm for generating the private keys. Each AC runs this algorithm independently to generate private keys for the attributes of the user it manages. The algorithm takes as input the global public key GP, the attribute set S_{AID} of the user governed by the independent AC, the system private key SK_{AID} governed by the independent AC, and the attribute set S_{user} submitted by the user. It generates the private key SK_{user}^{AID} corresponding to the attribute set of the user by the authorization center AID. Because we study the distributed CP-ABE in this paper, the attributes of each user may be managed by multiple authorization centers.

5) $SDecrypt(CT, SK_{user}) \rightarrow M$

The decryption algorithm. It takes as input the ciphertext CT and the attribute private key SK_{user} of the user. When the attributes of users satisfy the access policy tree corresponding to the ciphertext, the decryption algorithm outputs the plaintext M. Otherwise, it fails to decrypt.

6) $UPPolicy(S_{ap}, H, \mathcal{T}, Att_{\mathcal{T}}, C'_{\mathcal{T}}, R_l) \rightarrow (h, R_l, \mathcal{T}, \mathcal{T}', Att_{\mathcal{T}'}, C'_{R_l})$

The dynamic policy updating algorithm, which is run by the server. It takes as input the attribute policy set S_{ap}, the hash function H, the current policy $\mathcal{T}$, the attribute set $Att_{\mathcal{T}}$ contained in the current policy $\mathcal{T}$, the tuples $C'_{\mathcal{T}}$ (the same as C'_{R_l} below) in ciphertext C' corresponding to the attributes in the current policy $\mathcal{T}$, and the attribute revocation list R_l. It then generates $h, R_l, \mathcal{T}, \mathcal{T}', Att_{\mathcal{T}'}$, and C'_{R_l}, where $h = (h_1, h_1')$. h_1 is used to denote the attribute level of the current policy $\mathcal{T}$, and h_1' is used to indicate the attribute level of the updated policy $\mathcal{T}'$. $Att_{\mathcal{T}'}$ represents a set of attributes corresponding to the updated policy $\mathcal{T}'$, while C'_{R_l} denotes the tuples corresponding to the attributes in R_1 such as $(H(att_i^*))^{g^{\beta_i * r_i}}$, and they have been used to generate the original ciphertext C'. Here, the expression h_1 of the attribute set generated by the hash function H is used to ensure the security of the access policy and the correct policy mapping when decrypting. We consider only two kinds of demands for policy updating: (I) after the attributes are revoked, the new attribute set corresponds to new access policy already existing in the system, and the new access policy corresponds to a different privilege level from the old one (for the demand of the policy updating, we have stored the corresponding access policies for different privilege levels in the system and generate the random data r_i'' and $q_{\omega_i'}(0)$ for this updating); (II) the new attribute set does not change the privilege level of the users after the attributes are revoked. For example, revoking an attribute from the "OR" structure may not change the access privilege because the set of attributes that remain may still satisfy the policy tree with the same level of privilege. When attribute revocation affects the access privilege of users and there is no corresponding policy tree in policy set $S_{\mathcal{T}}$ after attribute revocation, we call this case the common attribute revocation; this case is beyond the scope of this paper and will be studied later.

7) *UPconSekey*$(GP, \{PK_{AID}\}, R_l^{AID}, Att_{\mathcal{T}'}^{AID}, h) \rightarrow (R_{UP}, W_{UP})$

The updating algorithm of the auxiliary function. It is reset by the data owner. It takes as input the global public key GP, the public keys $\{PK_{AID}\}$ issued by the ACs, the set R_l^{AID} of AID-managed attributes in R_l, the set $Att_{\mathcal{T}'}^{AID}$ of AID-managed attributes in $\mathcal{T}'$, and the parameter h representing the attribute level. Based on this input, it generates the parameters (R_{UP}, W_{UP}) that AID uses to grant the updated private key to the elements in the attribute revocation list R_l and (or) $\mathcal{T}'$. Here, the generated (R_{R_l}, W_{R_l}) in (R_{UP}, W_{UP}) for R_l^{AID} renders correct use of the revoked attribute parameters to decrypt the ciphertext impossible.

8) *UPKeyGen*$(GP, S_{AID}, SK_{AID}, h, R_l, Att_{\mathcal{T}'}, AID) \rightarrow SK_{user}^{AID'}$

The private key updating algorithm. This algorithm is run by the independent AC. It takes as input the global public key GP, the set S_{AID} of attributes governed by the AC AID, the private key SK_{AID} of the AC AID, h, the attribute revocation list R_l, the attribute set $Att_{\mathcal{T}'}$ that belongs to $\mathcal{T}'$, and the identity AID of the AC, and returns the updated private key $SK_{user}^{AID'}$ managed by AID.

9) *CTUpdate*$(CT, h, R_l, \mathcal{T}, \mathcal{T}', C'_{R_l}) \rightarrow CT'$

The ciphertext updating algorithm. This algorithm is run by the cloud server. It takes as input the previous ciphertext CT, h, the revocation list R_l, the original policy $\mathcal{T}$, the updated policy $\mathcal{T}'$, and the tuples C'_{R_l} corresponding to the attributes in R_1, and generates the new ciphertext CT' corresponding to the new access policy $\mathcal{T}'$.

C. SECURITY MODEL

Suppose that the cloud server is curious but honest, while the AC can be corrupted or compromised by an attacker. However, it is assumed that the attacker can corrupt the AC only statically, and the key query can be performed adaptively. In addition, we assume that the server may send owner data to users who do not have access privileges, but the data owner is completely trusted.

Users are assumed to be dishonest; for example, they may collude to access unauthorized data.

We describe the security model by presenting a game between the challenger and the attacker as follows:

Init. The attacker submits an access structure $\mathbb{A}^*$ to the challenger.

Setup. The attacker specifies a set of AC $S_A' \subset S_A$ that have been corrupted. The challenger generates the public/private key pairs by running the setup algorithms charged by the AC (suppose the global parameters GP have been given). For AC S_A-S_A' that are not corrupted, the challenger sends only their public keys to the attacker. For the AC S_A' that have been corrupted, the challenger sends their public and private keys to the attacker.

Phase 1. The attacker submits an attribute set $S = \{S_1, S_2, \ldots, S_{q_1}\}$ to the challenger for a private key query. The challenger generates the corresponding private key SK for the attacker. Here, S is required to not satisfy the access structure $\mathbb{A}^*$ submitted by the attacker; that is, the attacker does not obtain any key that can be used for decryption from these attributes, even if she/he combines effort with the AC that has been corrupted.

Challenge. The attacker submits two messages M_0, M_1 of equal length. The challenger randomly flips coin b, and encrypts M_b under the access structure $\mathbb{A}^*$, and then sends ciphertext CT^* to the attacker.

Phase 2. The attacker may query more private keys as long as the set $S_{q_1+1}, S_{q_1+2}, \ldots, S_{2q}$ of attributes provided by the attacker does not satisfy access structure $\mathbb{A}^*$. The attacker can also generate an update key query by submitting some attribute sets $S_{q_1'}, S_{q_2'}, \ldots, S_{q_n'}$, and the simulator returns the update key to the attacker.

Guess. The attacker outputs a guess b' of b.

In this game, the advantage of the attacker is defined as $P[b = b'] - 1/2$.

Definition 4: Suppose that the decisional q-BDHE assumption is true; then, the CP-ABE method proposed by us is safe only if all polynomial time attackers have at most a negligible advantage in the above security game.

V. OUR CONSTRUCTION OF CP-ABE

In this section, we present the construction of our improved CP-ABE and use $\mathcal{T}$ to represent the access policy based on the tree access structure used in this paper.

A. *GSetup*$(1^\lambda) \rightarrow GP$

The global setup algorithm for the CP-ABE system. This algorithm is run by a TTP. It takes as input the global security parameter 1^λ and chooses the p-order multiplicative cyclic group G_1, G_2, a bilinear map $e : G_1 \times G_1 \rightarrow G_2$, and the generator g of G_1. It also defines the Lagrangian coefficient $\Delta_{i,S}$ for an element set $S \subseteq Z_p$ in Z_p and $i \in Z_p$, such that $\Delta_{i,S}(x) = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$. At the same time, a hash function $H : \{0,1\}^* \rightarrow G_1$ is defined, and it is agreed that the hash value of the attribute att_i is $H(att_i)(i = 1, 2, \ldots, n)$ if the attribute of the user is not cut; if the attribute att_i is cut (called the negative attribute $\neg att_i$ in this paper), the hash value is $H(\neg att_j)(j = n+1, n+2, \ldots, 2n)$. In addition, this hash function is required to be strong and collision-free; even for the same attribute att_i, it must satisfy that $H(att_i)^r$ and $H(att_i)^{r'}$ are collision-free in G_1. Here $i = 1, 2, \ldots, n$, and n represents the size of the system-managed attribute set, and $r \neq r'$ are random numbers that are different from each other. Then, the global public key of the system is

$$GP = \{G_1, G_2, p, e, g, H, e(g, g)^\alpha\}$$

and the global master private key is $MK = \{\lambda, g^\alpha\}$.

B. AuthSetup$(GP, U_{AID}) \rightarrow (PK_{AID}, SK_{AID})$

The setup algorithm for the distributed AC. Each AC runs its own setup algorithm independently to generate its own public/private key pairs. Let U_{AID} represent the set of all attributes managed by the AC AID. For each attribute $att_i \in U_{AID}$, the AC selects the random number $\beta_i \in Z_p$ and issues the public key as follows:

$$PK_{AID} = \{g^{\beta_i}\}_{\forall att_i \in U_{AID}},$$

and the system private key of the AC is

$$SK_{AID} = \{\beta_i\}_{\forall att_i \in U_{AID}}.$$

C. SEncrypt$(GP, \{PK_{AID}\}, M, \mathcal{T}) \rightarrow CT$

the encryption algorithm. Suppose the set of leaf nodes contained in the policy tree $\mathcal{T}$ is Ω, and its modulus is N; then, there is

$$C = Me(g, g)^{\alpha s},$$
$$C' = \prod\nolimits_{i=1}^{N} (H(att_i^*))^{g^{\beta_i} * r_i}.$$

Therefore, $CT = \{\mathcal{T}, C, C'\}$. Here, s, $r_i \in Z_p(i = 1, 2, \ldots, N)$ are randomly selected by the encryptor (that is, the owner of image). att_i^* means that the attribute we want to calculate is either a positive attribute att_i or a negative attribute $\neg att_i$. Next, the data owner constructs a black box function $conSekey(GP, \{PK_{AID}\}, S_c, tag)$ shared with each independent AC. The function is used as the basic material by the AC to generate the attribute private key for the user. For this purpose, using the same method as in [1], we first select the polynomial q_x for each node x in the policy tree. Suppose that we select random number s for root node R_n and set $q_{R_n}(0) = s$. For other nodes, we set $q_x(0) = q_{parent(x)}(index(x))$. In addition, we choose the random number $q_{w_i}(0)$ for the leaf node ($\omega_i \in \Omega$ may be positive or negative attributes). Assume that the function $conSekey(GP, \{PK_{AID}\}, S_c, tag)$ has initialized the two-tuple—-the hash value $H(\omega_i)$ of each element in Ω and $q_{\omega_i}(0)$ corresponding to ω_i such as in [1]—-and randomly disturbed the order of the two-tuple. $|*|$ denotes the number of elements in set $*$ (the same as below). Then, the implementation process of $conSekey(GP, \{PK_{AID}\}, S_c, tag)$ is shown in Alg. 1.

Algorithm 1 $conSekey(GP, \{PK_{AID}\}, S_c, tag)$

Input: GP, $\{PK_{AID}\}$, the set S_c of attribute to be processed ($|S_c|= n_c$), and tag for the current attribute set of the user.

Output: A number set pair (R_{S_c}, W_{S_c}) corresponding to each element in S_c, h_1' generated when the attribute set of the user is calculated completely, and the additional parameter h_2.

```
$t_{slot}^{tag} = Time$. // $Time$ is the time slot set by the data owner for $tag$.
$T_{slot} = T_{slot} + t_{tag}$. //It is the sum of time $t_{tag}$ spent on all attributes tagged with $tag$
$Att = \{Att, S_c^{tag}\}$. //A new attribute set $S_c^{tag}$ is incorporated into the attribute set $Att$.
for $j$=1 to $n_c$ do
    for $i$=1 to $N$ do
        if $(S_c^j == \omega_i)$ then
            $R_{S_c^j} = H(\omega_i)^{g^{\beta_i} * r_i}$ //Here, $g^{\beta_i}$ and $r_i$ are the data selected above.
            $W_{S_c^j} = g^{\gamma * q_{\omega_i}(0)}$ //$\gamma \in Z_p$ is a random number chosen by the data owner.
        end if
    end for
end for
if $(T_{slot} == t_{slot}^{tag})$ then
    $h_1 = h_1' = H(p * rand())$ // This means that $h_1$ and $h_1'$ are the hash values of random numbers within $p$.
    $h_2 = g^{\frac{\alpha}{\gamma}}$
    return $\{(R_{S_c}, W_{S_c}), h_1', h_2\}$
else
    return $(R_{S_c}, W_{S_c})$
end if
```

We state that if the execution time of the algorithm for the tag exceeds the time slot, then each attribute corresponding to the tag is considered to have been generated with a private key such that the flag h_1' for the attribute level can be issued for the attribute set Att. Moreover, S_c^{tag} means the attribute set S_c marked by tag; h_1 is fed back to the data owner and saved on the TTP.

In this paper, to obtain $g^{\frac{\alpha}{\gamma}}$ but not leak the global master private key g^α, the image owner first sends $g^{\frac{1}{\gamma}}$ to the TTP through the secure channel, and then the TTP sends $g^{\frac{\alpha}{\gamma}}$ to the image owner, which is equivalent to the discrete logarithm problem, so it can be guaranteed that the value of g^α will not to be leaked to the image owner. Of course, α is also safe from any other third parties because γ is arbitrarily chosen by the image owner, and the image owner will not disclose his or her own data. h_2 can give each tag that calls $conSekey()$, or the tag that last calls $conSekey()$ because the value of h_2

remains unchanged. In this article, we use the latter method, and take h_2 as a global parameter.

In addition, we provide the following functions to determine the characteristics of attribute revocation, following the method in [5],

$$f(att_i^*) = \begin{cases} 1 & if\,(att_i^* \text{ } is\ a\ VA) \\ 0 & if\,(att_i^* \text{ } is\ not\ a\ VA). \end{cases} \tag{4}$$

The VA here represents a key attribute in the attribute set of the user. If a user attribute is a VA (that is, this attribute satisfies the condition set by the data owner for VA), then $f(att_i^*) = 1$, with no user revocation. In addition, this attribute may not be the identity(such as the ID number) of the user in the system. For example, at some point, the data owner may have the address as the VA.

D. $SKeyGen(GP, S_{AID}, SK_{AID}, S_{user}, AID) \rightarrow SK_{user}^{AID}$

the private key generation algorithm. Each authority center AC_i ($i = 1, 2, \ldots, w$, suppose a total of w ACs exist) needs to call this algorithm to generate private keys for users with the help of *conSekey*() as shown in Alg. 2. Here, *mod* denotes a modular function (the same as below). *tag* corresponds to S_{user}. τ_i is a random number selected by *AID*.

```
Algorithm 2 $SKeyGen(GP, S_{AID}, SK_{AID}, S_{user}, AID)$
Input: $GP, S_{AID}, SK_{AID}, S_{user}, AID$.
Output: The private key $SK_{user}^{AID}$ of the user.
[$R_{S_{AID}}, W_{S_{AID}}, h_1', h_2$] = $conSekey(GP, \{PK_{AID}\}, S_{AID}, tag)$
for $i$=1 to $|S_{user}|$ do
    for $j$=1 to $|S_{AID}|$ do
        if ($S_{user}^i == S_{AID}^j$ and mod($R_{S_{AID}}^j, H(S_{user}^i)^{g^{\beta_i}}$)==0) then
            $D_i = (W_{S_{AID}}^j)^{g^{\tau_i}} * R_{S_{AID}}^j$
            $D_i' = g^{-\tau_i}$
        end if
    end for
end for
return $SK_{user}^{AID} = \{h_1', h_2, D_i, D_i', i = 1, 2, \ldots, |S_{AID}|\}$
```

Therefore, the private key of user $SK_{user} = \{h_1', h_2, D, D'\}$, where D and D' are the set of D_i and D_i', respectively.

E. $SDecrypt(CT, SK_{user}) \rightarrow M$

the decryption algorithm. When decrypting, the value of the VA is first obtained by Eq.(4). If the value of the VA is 1, then we continue to perform the decryption operation; otherwise, the decryption algorithm terminates. During decryption, according to the policy tree, our method is the same as that used in [1]. If we let $\omega_s \in \Omega$, and r_s formally denotes a random number corresponding to the attribute ω_s, which may be r_1, r_2, or any other random number r_i, then the algorithm performs the following calculations recursively:

When x is a leaf node, we first calculate

$$\begin{aligned} &SDecryptNode(CT, SK_{user}, x) \\ &\quad = SDecryptNode(CT, SK_{user}, \omega_s) \\ &\quad = \frac{C'}{D_{\omega_s}} \\ &\quad = \frac{\prod_{i=1}^{N}(H(att_i^*))^{g^{\beta_i}*r_i}}{(W_{AID}^j)^{g^{\tau_i}} * R_{AID}^j} \\ &\quad = \frac{\prod_{i=1}^{N}(H(att_i^*))^{g^{\beta_i}*r_i}}{g^{\gamma*q_{\omega_s}(0)*g^{\tau_i}} * H(\omega_s)^{g^{\beta_i}*r_s}}. \end{aligned} \tag{5}$$

According to the encryption process and the private key generation process, when attribute ω_s of the user is a leaf node of the policy tree, the r_s in the denominator $H(\omega_s)^{g^{\beta_i}*r_s}$ in Eq.(5) is definitely an element r_i in the set of random numbers $\{r_1, r_2, \ldots, r_N\}$. In addition, we have identified the strong collision-free nature of the hash function when defining it. Therefore, it is perfectly possible to remove $H(\omega_s)^{g^{\beta_i}*r_s}$ from the numerator and denominator and not affect the numbers $H(att_i^*)^{g^{\beta_i}*r_i}$ related to other attributes, so that only $g^{\gamma*q_{\omega_s}(0)*g^{\tau_i}}$ is left in the denominator. Then, we compute

$$\begin{aligned} &(g^{\gamma*q_{\omega_s}(0)*g^{\tau_i}})^{D_i'} \\ &\quad = g^{\gamma*q_{\omega_s}(0)*g^{\tau_i}*g^{-\tau_i}} \\ &\quad = g^{\gamma*q_{\omega_s}(0)}. \end{aligned} \tag{6}$$

If x is not a leaf node, then

$$SDecryptNode(CT, SK_{user}, x) = \perp.$$

When a node x is a non-leaf node, for all child nodes c of x, it calls

$$SDecryptNode(CT, SK_{user}, c)$$

and saves the output as f_c. Let S_x be an arbitrary k_x-sized set of child nodes c such that $f_c \neq \perp$; then, the following can be computed by resorting to the Lagrange interpolation polynomial Eq.(1) and construction in *SEncrypt*():

$$\begin{aligned} f_x &= \prod_{c \in S_x} f_c^{\Delta_{i,S_x'}(0)} \\ &= \prod_{c \in S_x} (g^{\gamma*q_x(0)})^{\Delta_{i,S_x'}(0)} \\ &= \prod_{c \in S_x} (g^{\gamma*q_{parent(c)}(index(c))})^{\Delta_{i,S_x'}(0)} \\ &= \prod_{c \in S_x} (g^{\gamma*q_x(i)})^{\Delta_{i,S_x'}(0)} \\ &= g^{\gamma*q_x(0)}. \end{aligned}$$

Here, $i = index(c)$, $S_x' = \{index(c) : c \in S_x\}$. If the set of user attributes satisfies the policy tree, then we will ultimately obtain $g^{\gamma*s}$ when starting from the root node. Therefore, the user needs only to perform the following calculation to obtain the plaintext M,

$$\frac{C}{e(h_2, g^{\gamma*s})} = \frac{C}{e(g^{\frac{\alpha}{\gamma}}, g^{\gamma*s})} = \frac{C}{e(g, g)^{\alpha s}} = M.$$

F. $UPPolicy(S_{ap}, H, \mathcal{T}, Att_{\mathcal{T}}, C'_{\mathcal{T}}, R_l,) \rightarrow (h, R_l, \mathcal{T}, \mathcal{T}', Att_{\mathcal{T}'}, C'_{UP_{R_l}})$

the dynamic policy updating algorithm, which is run by the server. According to the type of attribute revocation provided in this paper, we consider only two policy update cases (I) and (II) proposed in Sec.IV-B; the implementation process as shown in Alg. 3.

Algorithm 3 $UPPolicy(S_{ap}, H, \mathcal{T}, Att_{\mathcal{T}}, C'_{\mathcal{T}}, R_l)$

```
Input: $S_{ap}, H, \mathcal{T}, Att_{\mathcal{T}}, C'_{\mathcal{T}}, R_l$.
Output: $h, R_l, \mathcal{T}, \mathcal{T}', Att_{\mathcal{T}'}, C'_{R_l}$.
$\mathcal{T}' = \mathcal{T} - R_l$ // This formally denotes that the attributes
  in attribute list $R_l$ are revoked from the current policy $\mathcal{T}$.

if ($\mathcal{T}' \in S_{ap}$) then
    kind=(I) // It means that the kind of revocation is (I).
    $h_1 = h_{\mathcal{T}}$
    $h'_1 = h_{\mathcal{T}'} = H(\mathcal{T}')$
    Obtain $Att_{\mathcal{T}'}$ based on $\mathcal{T}'$.
    Extract $C'_{R_l}$ from $C'_{\mathcal{T}}$ based on $R_l$.
end if
if ($\mathcal{T}' \notin S_{ap}$) then
    kind= (II) // It means that the kind of revocation is (II).

    $h_1 = h_{\mathcal{T}}$
    $h'_1 = h_T$
    $Att_{\mathcal{T}'} = \varnothing$ // $\varnothing$ means an empty set.
    Extract $C'_{R_l}$ from $C'_{\mathcal{T}}$ based on $R_l$.
end if
return $\{h, R_l, \mathcal{T}, \mathcal{T}', Att_{\mathcal{T}'}, C'_{R_l}\}$
```

Here, h, R_l, and $Att_{\mathcal{T}'}$ are given to *UPKeyGen*(), and h, R_l, $\mathcal{T}$, $\mathcal{T}'$, and C'_{R_l} are given to *CTUpdate*(). The hash value $h_{\mathcal{T}}$ of current policy $\mathcal{T}$ has been computed in Alg. 1, and the new value of h will be recorded by the data owner. In practical applications, for attribute security, some disturbing attributes are usually added into $Att_{\mathcal{T}'}$.

G. $UPconSekey(GP, \{PK_{AID}\}, R_l^{AID}, Att_{\mathcal{T}'}^{AID}, h) \rightarrow (R_{UP}, W_{UP})$

the auxiliary function updating algorithm. Let $(R_{R_l^j}, W_{R_l^j})$ and $(R_{Att_{\mathcal{T}'}^j}, W_{Att_{\mathcal{T}'}^j})$ represent the basic materials generated for the attributes in R_l^{AID} and $Att_{\mathcal{T}'}^{AID}$, respectively. This algorithm is run by the image owner, as shown in Alg. 4.

In the algorithm above, the attribute revocation would be the Type(I) revocation when $h_1 \neq h'_1$; otherwise, it is the Type (II) revocation. $Att_{\mathcal{T}'}^{AID}$ being a non-empty set means that the basic materials corresponding to it need to be updated, so $R_{Att_{\mathcal{T}'}}$ and $W_{Att_{\mathcal{T}'}}$ returned are not empty; otherwise, $R_{Att_{\mathcal{T}'}}$ and $W_{Att_{\mathcal{T}'}}$ are all empty sets. Moreover, g^{β_i} and r'_i have the same meanings as earlier, but $r'_i > r_i$ needs to be reselected to avoid having the same parameter value as the original one and, at the same time, ensure that $R_{R_l^j}$ cannot participate in the decryption calculation correctly. $\gamma', \gamma'' \in Z_p$ is the random number chosen by the data owner, and $q_{\omega'_i}(0)$ should be distinguished from its original random value for the specific attribute.

Algorithm 4 $UPconSekey(GP, \{PK_{AID}\}, R_l^{AID}, Att_{\mathcal{T}'}^{AID}, h)$

```
Input: $GP, \{PK_{AID}\}, R_l^{AID}, Att_{\mathcal{T}'}^{AID}, h$
Output: the number set pair $(R_{UP}, W_{UP})$
Obtain the leaf node sets $\Omega_{\mathcal{T}}$ and $\Omega_{\mathcal{T}'}$ of the policy trees
  corresponding to $h_1$ and $h'_1$ in $h$, respectively.
for $j$=1 to $|R_l^{AID}|$ do
    for $i$=1 to $|\Omega_{\mathcal{T}}|$ do
        if ($R_l^j == \omega_i$) then
            $R_{R_l^j} = H(\omega_i)^{g^{\beta_i * r'_i}}$// $R_l^j \in R_l^{AID}$
            $W_{R_l^j} = g^{\gamma' * q_{\omega_i}(0)}$// $\omega_i \in \Omega_{\mathcal{T}}$
        end if
    end for
end for
for $j$=1 to $|Att_{\mathcal{T}'}^{AID}|$ do
    for $i$=1 to $|\Omega_{\mathcal{T}'}|$ do
        if ($Att_{\mathcal{T}'}^j == \omega'_i$) then
            $R_{Att_{\mathcal{T}'}^j} = H(\omega'_i)^{g^{\beta_i * r''_i}}$ //$Att_{\mathcal{T}'}^j \in Att_{\mathcal{T}'}^{AID}$
            $W_{Att_{\mathcal{T}'}^j} = g^{\gamma'' * q_{\omega'_i}(0)}$// $\omega'_i \in \Omega_{\mathcal{T}'}$
        end if
    end for
end for
$R_{UP} = \{R_{Att_{\mathcal{T}'}}, R_{R_l}\}$
$W_{UP} = \{W_{Att_{\mathcal{T}'}}, W_{R_l}\}$
return $(R_{UP}, W_{UP})$
```

H. $UPKeyGen(GP, S_{AID}, SK_{AID}, h, R_l, Att_{\mathcal{T}'}, AID) \rightarrow SK_{user}^{AID'}$

the private key updating algorithm. Let D_{iR_l} and D'_{iR_l} represent the updated private keys generated for the i^{th} attribute in R_l^{AID}; then, the algorithm is shown in Alg. 5.

In addition, $Att_{\mathcal{T}'}^{AID}$ being a non-empty set means that the private keys corresponding to it need to be updated, so D and D' returned are not empty; otherwise, D and D' are all empty sets. Moreover, the new value of h'_1 will be returned to the user according to its original path.

According to Alg. 5, the private keys of the attributes that have not been revoked do not need to be updated when $h_1 = h'_1$.

I. $CTUpdate(CT, h, R_l, \mathcal{T}, \mathcal{T}', C'_{R_l}) \rightarrow CT'$

the ciphertext updating algorithm, and the implementation process of the algorithm is as follows:

If $h_1 \neq h'_1$, then let the data owner generate the new C and C' for the access policy $\mathcal{T}'$ corresponding to h'_1.

If $h_1 = h'_1$, then let

$$C' = \prod_{i=1}^{N} H(att_i^*)^{g^{\beta_i * r_i}} / \prod_{j=1}^{|R_l|} H(att_j^*)^{g^{\beta_j * r_j}}$$

based on the data $H(att_j^*)^{g^{\beta_j * r_j}}$ in C'_{R_l}, and let $\mathcal{T}' = \mathcal{T}$.

```
Algorithm 5 $UPKeyGen(GP, S_{AID}, SK_{AID}, h, R_l, Att_{\mathcal{T}'}, AID)$
Input: $GP, S_{AID}, SK_{AID}, h, R_l, Att_{\mathcal{T}'}, AID$.
Output: The updated private key $SK_{user}^{AID'}$ for users governed by $AID$.
Based on the AID, extract the attribute sets $R_l^{AID}$ and $Att_{\mathcal{T}'}^{AID}$ from $R_l$ and $Att_{\mathcal{T}'}$, respectively.
$[R_{UP}, W_{UP}] = UPconSekey(GP, \{PK_{AID}\}, R_l^{AID}, Att_{\mathcal{T}'}^{AID}, h)$.
if ($h_1 \neq h_1'$) then
    Extract $(R_{Att_{\mathcal{T}'}}, W_{Att_{\mathcal{T}'}})$ and $(R_{R_l}, W_{R_l})$ from $(R_{UP}, W_{UP})$
else
    Extract only $(R_{R_l}, W_{R_l})$ from $(R_{UP}, W_{UP})$.
end if
for $i$=1 to $|R_l^{AID}|$ do
    for $j$=1 to $|S_{AID}|$ do
        if (mod($R_{R_l}^i, H(S_{AID}^j)^{g^{\beta_i}}$)==0) then
            $D_{iR_l} = (W_{R_l}^i)^{g^{\tau_i'}} * R_{R_l}^i$
            $D_{iR_l}' = g^{-\tau_i'}$ //The value of random number $\tau_i'$ is different from $\tau_i$'s earlier.
        end if
    end for
end for
for $i = 1$ to $|Att_{\mathcal{T}'}^{AID}|$ do
    for $j$=1 to $|S_{AID}|$ do
        if (mod($R_{Att_{\mathcal{T}'}}^i, H(S_{AID}^j)^{g^{\beta_i}}$)==0) then
            $D_i = (W_{Att_{\mathcal{T}'}}^i)^{g^{\tau_i^u}} * R_{Att_{\mathcal{T}'}}^i$
            $D_i' = g^{-\tau_i^u}$
        end if
    end for
end for
return $SK_{user}^{AID'} = \{(D_i, D_i'), (D_{iR_l}, D_{iR_l}')\}$
```

Therefore, the output of this algorithm is $CT' = \{\mathcal{T}', C, C'\}$.

J. REVOCATION

In this paper, we use the VA to achieve user revocation. Before decryption, we first determine whether to revoke the user based on the value of the VA. For attribute revocation, in this paper, we study two types of revocation corresponding to the two cases discussed in the policy update algorithm: (i) attribute revocation causes the privilege level of the user to be reduced to a lower level; (ii) attribute revocation simply revokes some attributes that have no effect on the privilege of the user.

The algorithm implementation process when the attribute is revoked is: first, the data owner initiates the attribute revocation request by passing R_l, $\mathcal{T}$, and $C_{\mathcal{T}}'$ to the server, and then the server executes the algorithm *UPPolicy*() to complete the policy update; next, the algorithms *UPKeyGen*() and *CTUpdate*() complete the private key update and the ciphertext update. It can be seen from the implementation process of the policy update algorithm that for the (i)-type revocation, we must issue a new private key to the user; while for the (ii)-type revocation, using the algorithm provided in this article, only the affected attribute's private key must be updated so that it cannot participate in the calculation correctly.

VI. PROOF OF THE CORRECTNESS AND SECURITY OF OUR PROPOSED CP-ABE

A. PROOF OF CORRECTNESS

Theorem 1: Our CP-ABE scheme is correct.

Proof: First, from the implementation process for the decryption algorithm before updating, when an attribute is a leaf node, we can calculate $g^{\gamma * q_{\omega_s}(0)}$. When a node x is a non-leaf node, based on the Lagrange interpolation polynomial Eq.(1) and the construction in *SEncrypt*(), $g^{\gamma * q_x(0)}$ for the node x can also be calculated on all the child nodes c of x. In a similar fashion, when the attribute set of the user satisfies the policy tree, the $g^{\gamma * s}$ for the root node can finally be calculated. Then, we can perform the bilinear pairing calculations as follows:

$$e(h_2, g^{\gamma * s}) = e(g^{\frac{\alpha}{\gamma}}, g^{\gamma * s}) = e(g, g)^{\alpha * s}.$$

Therefore,

$$\frac{C}{e(g, g)^{\alpha * s}} = \frac{M * e(g, g)^{\alpha * s}}{e(g, g)^{\alpha * s}} = M.$$

Second, from the updated ciphertext structure, policy tree structure, and private key composition, regardless of the Type (I) update or the Type (II) update, the judgment regarding the privilege level of the user can be implemented based on the non-revoked attributes, and these attributes enable the corresponding $g^{\gamma * q_{\omega_s}(0)}$ to be obtained from calculation of the policy tree by Eq.(5) and (6), so that the correct decryption operation can be achieved.

In conclusion, our proposed CP-ABE scheme is correct. ■

B. PROOF FOR SECURITY

To realize the distributed attribute-based encryption, in this paper, we set a TTP to generate the GP. Each AC then uses the GP to independently generate the corresponding private keys for the attributes they are responsible for. At the same time, the data owner encrypts the plaintext using the GP and the public keys issued by the AC. It should be noted that the TTP designed in this paper is not responsible for the authorization. It is responsible for determining the accessor privilege before formal decryption and for the setup of the master system parameters.

As mentioned earlier, the ACs are honest but curious; they do not actively attack the received data but simply view it. In addition, they are attacked only statically, that is, they will not actively cooperate with adversaries to attack the system before being corrupted. Suppose the adversary can attack the AC, thereby obtaining the private key SK_{AID} issued by the AC having identity AID. If the adversary breaks through some AC, the centers will send the data returned by *conSekey*()

to the adversary. Based on this scenario, we provide Theorem 2 and its proof process as follows.

Theorem 2: Suppose that the decisional q-BDHE assumption is true. Then, no adversary can selectively break through our scheme in polynomial time with a non-negligible advantage.

Proof: Assuming that adversary $\mathcal{A}$ can win the security game with the advantage of ε, then we can construct a challenger *Cha* to solve the decisional q-BDHE problem with a non-negligible advantage $\varepsilon/2$, thus distinguishing $e(g, g)^{a^{q+1}s}$ and a random element in G_2. The security game between *Cha* and $\mathcal{A}$ is as follows:

Suppose the challenger *Cha* inputs a q-BDHE instance $Y_{g,a,q}$ and Z_e, either $Z_e = e(g, g)^{a^{q+1}s}$, or Z_e is a random number in G_2.

Without loss of generality, we assume that only AC_1 is not corrupted by adversary $\mathcal{A}$. Before the game starts, $\mathcal{A}$ submits a challenge access policy tree $\mathcal{T}^c$ to the challenger. There are $|\Omega_c|$ attributes in the policy tree $\mathcal{T}^c$, and they satisfy $\Omega^* = U_1 \bigcap \Omega_c \neq \varnothing$. The space size of system attributes is $|U| = NN$.

1) SETUP

Cha randomly selects $i^* \in \{i|att_i \in \Omega^*\}$ and a random number $\beta'_i \in Z_p, i = 1, 2, \ldots, NN$. *Cha* then calculates the following numbers.

First, for the public parameter $e(g, g)^\alpha = e(g^{a^q}, g^a) * e(g, g)^{\alpha'}$, *Cha* confidentially sets $\alpha = \alpha' + a^{q+1}$. At the same time, *Cha* also confidentially sets $\beta_{i^*} = \beta'_{i^*} + \Sigma_{att_j \in \Omega^*, j \neq i^*} a^{q+1-j}$.

Then *Cha* computes

$$g^{\beta_i} = g^{\beta_{i^*}} = g^{\beta'_{i^*}} \prod_{att_j \in \Omega^*, j \neq i^*} g_{q+1-j}, \tag{7}$$

where we set $g_{q+1-j} = g^{a^{q+1-j}}$ the same as below.

For $i \in \{i|att_i \in \Omega^*/\{att_i^*\}\}$, calculate

$$g^{\beta_i} = g^{\beta'_i} g_{q+1-i}^{-1}.$$

In other cases, *Cha* calculates $g^{\beta_i} = g^{\beta'_i}$.

Moreover, to construct the auxiliary function, *Cha* randomly generates $\{r'_i, i = 1, 2, \ldots NN\}$ and generates the corresponding polynomials for each node in the tree according to the access policy tree challenged by the adversary, and finally generates $\{q'_{\omega_y}(0), y = 1, 2, \ldots, |\Omega_c|\}$ for the leaf node set. Assuming that the set of attributes to be processed is S'_c, then *Cha* can construct the auxiliary function *SPconSekey*() as shown in Alg. 6.

Algorithm 6 *SPconSekey*(GP, $\{PK_{AID}\}$, S'_c)

Input: GP, $\{PK_{AID}\}$, and the attribute set S'_c, where $|S'_c| = t$.
Output: h_2, and the number set pairs $(R_{S'_c}, W_{S'_c})$ corresponding to the elements in S'_c.

```
for $j$=1 to $t$ do
    for $y$=1 to $|\Omega_c|$ do
        if $((S'_c)_j == \omega'_y)$ then
            $R_{(S'_c)_j} = H(\omega'_y)^{g^{\beta_i * r_i^c}}$ //$g^{\beta_i}$ and $r_i^c$ have the same meaning as in Alg. 1.
            $W_{(S'_c)_j} = g^{\gamma^c * q^c_{\omega_y}(0)}$
        end if
    end for
end for
$h_2 = g^{\frac{\alpha}{\gamma}}$
return $\{(R_{S'_c}, W_{S'_c}), h_2\}$
```

where g^{β_i} is the same as Eq.(7) when $i \in \{i|att_i \in \Omega^*\}$; and

$$r_i^c = r_i^{c'} * \prod_{att_j \in \Omega^*, j \neq i^*} g_{q+1-j},$$

$$q^c_{\omega_y}(0) = q^{c'}_{\omega_y}(0) + \sum_{att_j \in \Omega^*, j \neq i^*} a^{q+1-j},$$

$$\tau_i^c = \tau_i^{c'} + \sum_{att_j \in \Omega^*, j \neq i^*} a^{q+1-j},$$

$$\gamma^c = \gamma^{c'} + \sum_{att_j \in \Omega^*, j \neq i^*} a^{q+1-j}.$$

For $i \in \{i|att_i \in \Omega^*/\{att_i^*\}\}$, set

$$\begin{aligned} r_i^c &= r_i^{c'} * g_{q+1-j}, \\ q^c_{\omega_y}(0) &= q^{c'}_{\omega_y}(0) + a^{q+1-j}, \\ \tau_i^c &= \tau_i^{c'} + a^{q+1-j}, \\ \gamma^c &= \gamma^{c'} + a^{q+1-j}. \end{aligned}$$

For the other i, set $r_i^c = r_i^{c'}$, $q^c_{\omega_y}(0) = q^{c'}_{\omega_y}(0)$, $\tau_i^c = \tau_i^{c'}$, $\gamma^c = \gamma^{c'}$. Here, c is used to indicate that these data are related to $\mathcal{T}^c$.

The challenger sends $(p, G_1, G_2, e, Y_{g,a,q})$, $\{g^{\beta_i}|att_i \in U\}$, $\{\tau_i^c, \beta_i|att_i \in U/U_1\}$ even $\{(R_{S'_c}, W_{S'_c}), h_2\}$ to adversary $\mathcal{A}$ (assuming that these data correspond to the corrupted ACs, but the data belonging to the uncorrupted authorization center AC_1 are not sent to adversary $\mathcal{A}$).

2) PHASE 1

Adversary $\mathcal{A}$ can adaptively implement any key query to *Cha* with any set Ω' of attributes and requires only that the attribute set Ω' does not satisfy the policy tree $\mathcal{T}^c$ challenged by $\mathcal{A}$.

If $i = i^*$,

$$\begin{aligned} D_i^* &= (W^j_{AID})^{g^{\tau_i^*}} * R^j_{AID} \\ &= g^{q'_{\omega_i}(0) * \gamma' * g^{\tau'_i}} * (\prod_{att_j \in \Omega^*, j \neq i^*} g_{q+1-j})^3 \\ &\quad * H(\omega_j^*)^{g^{\beta_i^* + 2\sum_{att_j \in \Omega^*, j \neq i^*} a^{q+1-j}} * r'_i}, \\ D_i^{*'} &= g^{-\tau_i} = g^{-\tau'_i} * \prod_{att_j \in \Omega^*, j \neq i^*} g^{-1}_{q+1-j}. \end{aligned}$$

For $i \neq i^*$, $att_i' \in \Omega^*/\{i = i^*\}$,

$$\begin{aligned} D_i &= (W_{AID}^j)^{g^{\tau_i^*}} * R_{AID}^j \\ &= g^{\gamma' * q'_{\omega_i}(0) * g^{\tau_i'}} (g_{q+1-j})^3 \\ &\quad * H(\omega_j^*)^{g^{\beta_i'} * r_i' * (g_{a+1-j})^2}, \\ D_i' &= g^{-\tau_i} = g^{-\tau_i'} * g_{q+1-j}^{-1}. \end{aligned}$$

For $i \neq i^*$, and $att_i' \in U/U_1$,

$$\begin{aligned} D_i &= (W_{AID}^j)^{g^{\tau_i}} * R_{AID}^j \\ &= g^{q'_{\omega_i}(0) * \gamma' * g^{\tau_i'}} * H(\omega_j^*)^{g^{\beta_i'} * r_i'}, \\ D_i' &= g^{-\tau_i} = g^{-\tau_i'}. \end{aligned}$$

Here, τ_i', γ', and $q'_{\omega_i}(0)$ are randomly selected by *Cha*; they are related to Ω' and let private keys meet the requirements of random distribution; and they can be calculated.

3) CHALLENGE

At some point, adversary $\mathcal{A}$ submits to *Cha* two equal-length messages M_0 and M_1, and *Cha* randomly selects $b \in \{0, 1\}$ to calculate:

$$\begin{aligned} C &= M_b * e(g, g)^{\alpha s} = M_b * e(g^{\alpha}, g^s) \\ &= M_b * e(g^{a^{q+1}} * g^{\alpha'}, g^s) \\ &= M_b * e(g, g)^{a^{q+1} s} * e(g, g)^{\alpha' s} \\ C' &= \prod_{att_i \in \Omega} H(\omega_j)^{g^{\beta_i} * r_i} \\ &= ((H(\omega_j)^{g^{\beta_i'}} * \prod_{att_j \in \Omega^*, j \neq i^*} H(\omega_j)^{2g^{q+1-j}})^{r_j'} \\ &\quad * (H(\omega_{i'})^{g^{\beta_{i'}}} * \prod_{att_i \in \Omega^*, i \neq i^*} H(\omega_{i'})^{2g_{q+1-i}})^{r_i'} \\ &\quad * \prod_{att_i \in \Omega / \Omega^*} H(\omega_{i'})^{g^{\beta_i'} * g^{r_i'}}). \end{aligned}$$

when $Z_e = e(g, g)^{a^{q+1} s}$, C is the legal ciphertext of plaintext M_b. Otherwise, in the eyes of the adversary, C is a random number of G_2.

4) PHASE 2

Same as Phase 1.

5) GUESS

The adversary $\mathcal{A}$ outputs the guess value b' of b. If $b' = b$, *Cha* outputs 0; otherwise, the output is 1.

Analysis:

1) When *Cha* outputs 0, C is a legal ciphertext. In this case, the adversary can exert its full attack advantage. Assuming that the attack advantage of adversary $\mathcal{A}$ is $\varepsilon = P[b = b'] - 1/2$, it is easy to see that $P[b = b'] = P[b = b' | b = 0]$, so when $b = 0$, the probability that *Cha* wins is

$$P[b = b' | b = 0] = P[b = b'] = \varepsilon + 1/2.$$

2) When $b = 1$, Z_e is an element randomly selected from G_2, so for the adversary $\mathcal{A}$, C is only a random element in G_2 and does not contain any message of plaintext M_b. In this case, $\mathcal{A}$ loses its attack advantage. Thus, $P[b \neq b'] = 1/2$. It is easy to see that $P[b \neq b'] = P[b = b' | b = 1]$. Therefore, when $b = 1$, the probability that *Cha* wins is

$$P[b' \neq b] = P[b = b' | b = 1] = 1/2.$$

Finally, *Cha's* advantage in solving the decisional q-BDHE hypothesis is as follows:

$$\begin{aligned} P[b = b'] - 1/2 &= P[b = b' | b = 0] * P[b = 0] \\ &\quad + P[b = b' | b = 1] * P[b = 1] - 1/2 \\ &= (\varepsilon + 1/2) * 1/2 + 1/2 * 1/2 - 1/2 = \varepsilon/2. \end{aligned}$$

Therefore, if the adversary can break through the encryption scheme above with the advantage of ε, then the challenger can solve the decisional q-BDHE problem with the advantage of $\varepsilon/2$. Obviously, this is impossible, so the assumption is not true. Thus, no adversary can selectively break through our scheme in polynomial time with a non-negligible advantage. ■

VII. A PROGRESSIVELY CLEAR IMAGE RESTORATION ALGORITHM BASED ON CP-ABE

A distributed CP-ABE with a shorter constant-size ciphertext and only one bilinear pairing calculation is provided above, and its correctness and security are proven. This opens up the possibilities for its application on smart terminals with limited storage and calculation power. In this section, we will provide an image progressive restoration algorithm based on our CP-ABE. For this purpose, we first provide the algorithm for restoring images with reference to the atmospheric scattering model, and then provide an image restoration algorithm based on our proposed CP-ABE and the algorithms for restoring images.

A. THE ALGORITHM FOR PROGRESSIVELY AND CLEARLY RESTORING AN IMAGE BASED ON THE ATMOSPHERIC SCATTERING MODEL

Although the algorithms proposed in [34] (denoising), [35]–[37] (image restoration), [38] (inpainting), and [39] (dehazing) have high restoring performance, none of these algorithms can restore the images with progressive clarity. Moreover, it is known from related work that some studies provide algorithms that can achieve progressively clear image restoration, but these algorithms still require further improvement in the peak signal to noise ratio (PSNR) of the ultimately restored image, especially when the original clear image needs to be restored; for example, medical images require lossless restoration for a correct diagnosis. For the methods of lossless restoration, the algorithms in [11]–[13] are not the best choice due to their high storage costs, even if they can completely restore the original image. The algorithm in [14] means that the encrypted image (as the carrier) is not restored completely; and the hiding rate (bpp) of this type of algorithm is very low; therefore it is not favorable for the restoration of large-size images.

Therefore, in this section, we provide three algorithms that rely only on the image itself for progressive restoration of the image in dehazing mode with reference to the atmospheric scattering model: the loss progressively clear image restoration algorithm, the lossless progressively clear image restoration algorithm, and the reversible convolution transformation (RCT) plus lossless progressively clear image restoration algorithm.

1) THE LOSS PROGRESSIVELY CLEAR IMAGE RESTORATION ALGORITHM

We first provide the algorithm for loss progressively clear image restoration. The principle of this algorithm is as follows: first, the original image is hazed with a given density of haze based on Eq.(2); then, for the hazed image, the inverse process of hazing is used to restore the image with progressive clarity. The specific restoration process is shown in Alg. 7.

Algorithm 7 The Loss Progressively Clear Image Restoration Algorithm

```
Input: The hazed color image data $I$, parameters $\triangle_{\infty}$, $\varphi$,
    the formula for the depth of field $\zeta(i,j)$, and the attribute
    level $L'_{att}$ of the user.
Output: The image data $I_{L'_{att}}$ with sharpness corresponding
    to the attribute level of the user.
for $l$=1 to 3 do
    for $i$=1 to $I_{row}$ do
        for $j$=1 to $I_{column}$ do
            $\xi(i,j) = e^{-\varphi * \zeta(i,j)}$
            $I_{L'_{att}}(i,j,l) = (J(i,j,l) - \Delta_{\infty} * (1 - \xi(i,j)))/\xi(i,j)$
        end for
    end for
end for
```

It can be seen from Eq.(3) that the value of $\xi(i,j)$ is a floating-point number, and then the image hazing and restoring processes described above are both lost; therefore, Alg. 7 is mainly applied to the progressively clear restoration of the image with a loss format such as jpg.

2) THE LOSSLESS PROGRESSIVELY CLEAR IMAGE RESTORATION ALGORITHM

There are some cases that require the lossless restoration of the image, such as medical images. To this end, we provide a lossless progressively clear image restoration algorithm using the atmospheric scattering model. Here, based on Eq.(2), we propose Eq.(8) to generate a multilayer scrambled image as the basic data for lossless progressively clear image restoration.

$$I(i,j) = mod(J(i,j) - \Delta_{\infty} * L_{att} + \varphi * (\mathcal{G} - \xi(i,j)), 256), \tag{8}$$

where $\xi(i,j) = mod(2^{mod(i+j,\phi_1)} * \varphi, \phi_2)$. L_{att} indicates the disturbance intensity, which corresponds to the attribute level of the user; $\mathcal{G}$, ϕ_1 and ϕ_2 can all be used to control the disturbance intensity of the image. We add these factors to the atmospheric scattering model. The meanings of the remaining parameters are the same as in the earlier atmospheric scattering model. Since we are implementing distortion-free image restoration, the parameters used here are all integers.

Given the above scrambling formula, we provide Alg. 8 to progressively restore the protected images without distortion. Here, $\xi(i,j)_{L_{att}}$ represents the value of $\xi(i,j)$ corresponding to the scrambling level L_{att}, and $L_{largest}$ is the highest attribute level.

Algorithm 8 The Lossless Progressively Clear Image Restoration Algorithm

```
Input: The protected color image data $I$, parameters $\triangle_{\infty}$, $\varphi$,
    $\mathcal{G}$, $\phi_1$ and $\phi_2$, and the attribute level $L'_{att}$ of the user.
Output: The image data $I_{L'_{att}}$ with sharpness corresponding
    to the attribute level of the user.
for $l$=1 to 3 do
    for $i$=1 to $I_{row}$ do
        for $j$=1 to $I_{column}$ do
            $\xi(i,j) = mod(2^{mod(i+j,\phi_1)} * \varphi, \phi_2)$
            $I_{L'_{att}}(i,j,l) = mod(J(i,j,l) + \Delta_{\infty} * L'_{att} + (L_{largest} - \varphi) * (\mathcal{G} - \xi(i,j)) - L_{att} * (\mathcal{G} - \xi(i,j)_{L_{att}})), 256)$
        end for
    end for
end for
```

3) THE RCT PLUS LOSSLESS PROGRESSIVELY CLEAR IMAGE RESTORATION ALGORITHM

Due to the strong restoring capacity of the existing image restoration algorithms, if only one method is used to protect image information, then the image remains extremely vulnerable to image restoration attacks. Therefore, we provide a novel algorithm in this section—the RCT plus lossless progressively clear image restoration algorithm. In this algorithm, the RCT is first performed on the original image data. Then, the transformed data are scrambled, and the scrambled image data are finally used as the basic data for the progressively clear restoration of the image. Its implementation process is shown in Alg. 9.

It is worth noting that in the process of implementing the algorithm, to ensure that the image after convolution has a certain degree of readability, the selected convolution kernel should minimize change to the original image data, and simultaneously, it should be implemented accurately. Therefore, the selection of a convolution kernel is very important. The convolution kernel used in this experiment is a two-dimensional matrix (1 0; 1 1), and it is used as part of the image sharpness parameter. Moreover, the process of obtaining the basic data for progressively

Algorithm 9 The RCT Plus Lossless Progressively Clear Image Restoration Algorithm

```
Input: The parameters $\triangle_\infty, \varphi, \mathcal{G}, \phi_1$ and $\phi_2$, and the attribute
    level $L'_{att}$ of the user. The protected color image data $I$
    after convolutional transformation.
Output: The image data $I_{L'_{att}}$ with sharpness corresponding
    to the attribute level of the user.
for $l$=1 to 3 do
    for $i$=1 to $I_{row}$ do
        for $j$=1 to $I_{column}$ do
            $\xi(i,j) = mod(2^{mod(i+j,\phi_1)} * \varphi, \phi_2)$
            $I_{L'_{att}}(i,j,l) = mod(J(i,j,l) + \Delta_\infty * L'_{att} + (L_{largest} - \varphi) * (\mathcal{G} - \xi(i,j)) - L_{att} * (\mathcal{G} - \xi(i,j)_{L_{att}})), 256)$
        end for
    end for
end for
If the original image needs to be restored, then a deconvolution transform is performed; otherwise, the algorithm
    terminates.
```

Algorithm 10 The Progressively Clear Image Restoration Algorithm Based on Our CP-ABE Scheme and the Algorithms for Restoring Images

```
Input: The private key $SK_{user}$ of the user, the ciphertext $CT$,
    and the protected image with our algorithms.
Output: An image whose sharpness matches the attribute
    level of the user.
Obtain parameter $h'_1$ from $SK_{user}$ and match it with $h_1$
    held by the TTP.
Obtain the corresponding policy according to the
    attribute set to which $(h_1, h'_1)$ belongs, and then obtain
    the ciphertext encrypted by this policy.
Decrypt the ciphertext with the user's private key $SK_{user}$
    and obtain the image sharpness parameters (such as
    $\Delta_\infty$, $\varphi$, and the convolution kernel) corresponding to the
    attribute level of the user.
if the image sharpness parameters are all floating-point
    numbers then
    Call Alg. 7 to restore the image.
end if
if the image sharpness parameters are all the integer
    numbers then
    if there is no convolution kernel in the parameters then
        Call Alg. 8 to restore the image.
    else
        Call Alg. 9 to restore the image.
    end if
end if
```

clear image restoration is the same as that for Alg. 8 after convolution.

Given the current state of the art, no image restoration (or repairing) algorithm can completely restore the disturbed image without the original image data. In this way, even if an image restoration algorithm can restore a higher sharpness image and can guess the convolution kernel used, it cannot completely recover the original image used in the algorithm. Because an attack on lossless scrambling does not completely restore the disturbed image data to the original state after convolution, the attacker can implement inverse convolution only on the wrong data. Thus, this image protection method using multilevel scrambling is better at protecting image information security.

B. THE PROGRESSIVELY CLEAR IMAGE RESTORATION ALGORITHM USING OUR CP-ABE

In this section, we present a progressively clear image restoration algorithm based on our proposed CP-ABE and the algorithms for restoring images, as shown in Alg. 10. Here, the plaintext encrypted by CP-ABE is the image sharpness parameter, which is used to restore the protected image to the corresponding sharpness.

Obviously, as the policy level corresponding to the attribute set of the user increases, the image he or she receives will become increasingly clear.

VIII. EXPERIMENT AND DISCUSSION

The computer used in our experiments is equipped with Win 10 OS and an Intel(R) Celer-on(R) CPU with a frequency of 2.81 GHz. The algorithm runs on Java 7. In the experiments, we use 600 policies, each of which corresponds to 10 plaintexts. Additionally, the experiments are carried out on 200 images (100 jpg images and 100 png images), each of which is 256*256 in size. On the whole, we implement more than 6000 experiments about the loss and lossless restoration. During the experiment, we first test the performance of the proposed algorithm by the time cost. Then, based on the restored image, we examine the image restoration capability of the proposed algorithm. Finally, we analyze the advantages of our algorithms by comparison with other algorithms on the quality of image restoration, the size of ciphertext, and the number of bilinear pairing calculations. Detailed experimental results are given below.

A. DEMONSTRATION OF THE PERFORMANCE OF THE IMPROVED CP-ABE ALGORITHM

In this section, we demonstrate the performance of the improved CP-ABE algorithm by verifying two time costs: encryption time and decryption time. In these experiments, we change the attribute combinations to reflect the same level but different policies. In addition, when transforming attribute combinations, we use the positive and negative attributes in each attribute combination. For different policy levels, we change the number of attributes on the one hand, and change the number of operation types on the other hand, thereby changing the complexity of the attribute policy. Furthermore, we emphasize the following considerations.

Our improvements are based on the CP-ABE in [1], and the algorithms in [40], [41] are all the improved algorithms of CP-ABE; at the same time, the scheme in [40] had the idea of hierarchical access control, the scheme in [41] discussed the constant-size ciphertext and the distributed CP-ABE, and these are basically the same as the idea in this paper. Therefore, we here compare the two average time costs with the schemes in [1], [40], [41] to illustrate the good performance of our scheme as shown below.

1) ENCRYPTION TIME VERIFICATION

In this experiment, we verify the cost of the encryption time using 6 levels of image sharpness. In addition, we design 100 different policies for each level of sharpness. Then, the encryptions for 10 different plaintexts are performed for each policy. The experimental results are shown in Fig. 2.

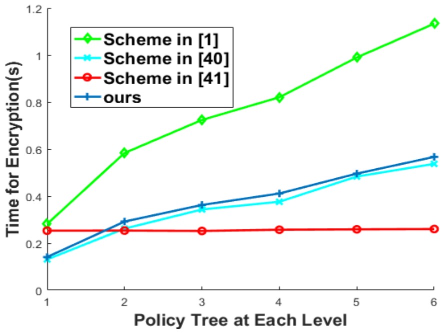


FIGURE 2. **The experimental results of encryption time.**

It can be seen from Fig. 2 that our encryption time cost is smaller than that provided in [1], generally higher than that provided in [41], and equivalent to that provided in [40].

2) DECRYPTION TIME VERIFICATION

We complete this experiment using the same scheme as above. The experimental results are shown in Fig. 3.

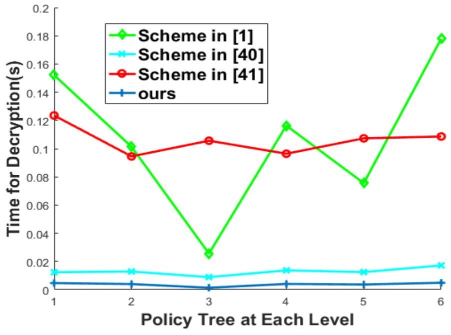


FIGURE 3. **The experimental results of decryption time.**

As can be seen from Fig. 3, the decryption time of our improved CP-ABE is less than the time cost of the algorithms provided in [1], [40], [41].

Notably, in the experiment, the numbers of attributes corresponding to the six image sharpness levels we selected are 5, 10, 14, 16, 20, and 23, respectively. The plaintext encrypted by the CP-ABE algorithm is the image sharpness parameters, which contain approximately 30 float data. However, we found that when the composition of the policy is not complicated, even though it contains more attributes, the time spent on decryption calculation will be lower. The time costs of level 3 and 5 in Fig. 3 are lower than that of level 1 for this reason.

In a word, as shown in Figs. 2 and 3, the advantages in time cost of our proposed CP-ABE can be seen from more than 6,000 experimental results.

B. DEMONSTRATION OF THE IMAGE RESTORATION PERFORMANCE OF OUR STRATEGY

In this section, we demonstrate the performance of our strategy in progressively clear restoration of images by presenting practical restored images. In the VIII-A experiments, for convenient observation, only 6 levels for progressively clear restoring are designed. We then use 6 levels of the attribute policy to correspond to the 6 levels of clarity. That is, if the attribute set of the user satisfies the first attribute level (the lowest attribute level), then a restoration result of clarity level 1 for the image is shown to the accessor; if the attribute set of the user meets the sixth attribute level (the highest level), then a restoration result of clarity level 6 (original image) is shown to the accessor. We use 100 images to correspond to the 100 strategies in each level, and the plaintexts under the same privilege level correspond to different combinations of the sharpness parameters. We perform the experiments in two classes: overall image restoration and restoration of the hazed ROIs in the image, and verify only the ROI restoration in the loss situation because the lossless ROI restorations are similar to the loss restoration. For the three algorithms adopted in this paper, we provide the corresponding experiments as follows.

1) LOSS PROGRESSIVE RESTORATION FOR THE IMAGE

In this section, we present only the test results corresponding to these parameters $\Delta_\infty = 0.8$, $\varphi = [0, 0.05, 0.10, 0.15, 0.20, 0.25]$, and $\zeta(i,j) = -0.05 * j + 20$, and we show the four images out of 100 jpg images as represented in Fig. 4, where "in(output 1)" means the input image and its sharpness is the lowest; at the same time, this image is also the output image of level 1. "output i" represents the image restored by Alg.7; here "i" corresponds to the user's attribute level. The larger the value of "i", the higher the level is, and the sharpness of the restored image is also higher. In the experiment, we use the input image as the image restored for the user with the lowest privilege level (the same is true below).

As shown in Fig. 4, the algorithm we provide restores clearly and progressively the protected images.

In Fig. 5, the size of the selected ROIs is $64 * 64$, and the values of other parameters are the same as those in Fig. 4. It can be seen from Figs. 4 and 5 that, with our provided algorithms, we can protect both the entirety and parts of

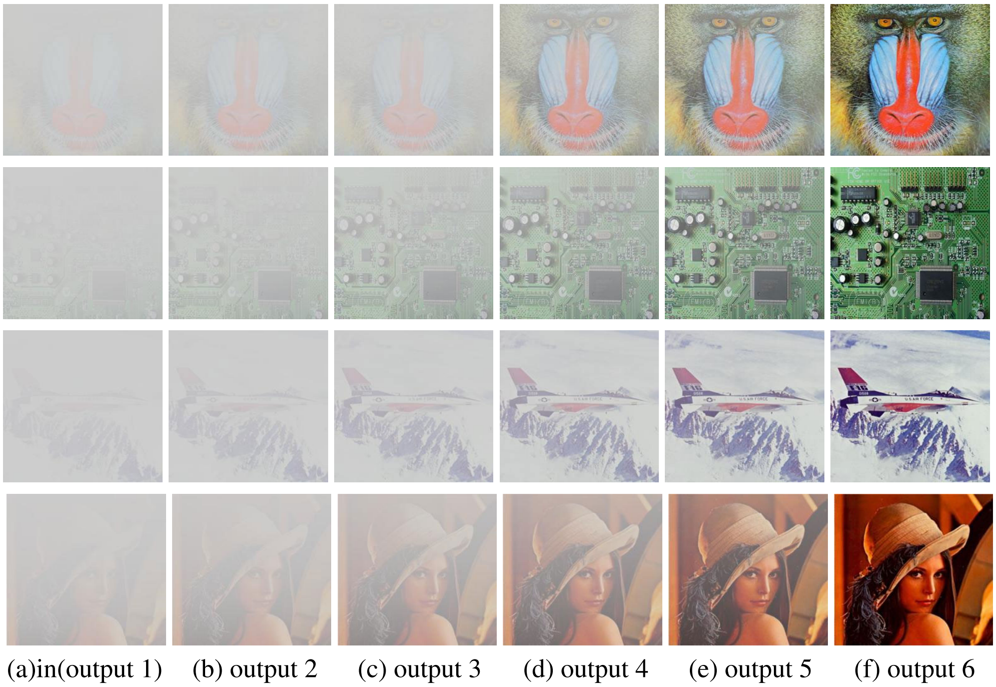


FIGURE 4. Overall image restoration results for 6 levels.

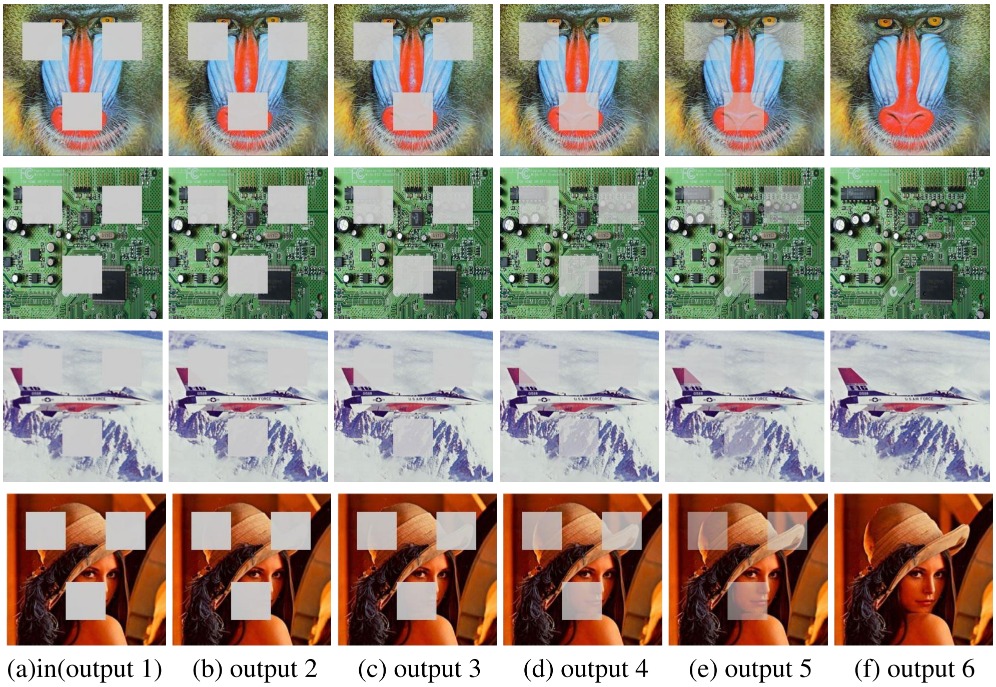


FIGURE 5. The image restoration results of 6 levels for ROIs in the image.

an image. Moreover, the fully restored images have high PSNR relative to the original clear images (see Table 2).

2) LOSSLESS PROGRESSIVELY CLEAR IMAGE RESTORATION

In this section, we present only the experimental results of the full image progressive restoration, as shown in Fig. 6. The sharpness parameters are $\Delta_\infty = [0, 2, 4, 6, 8, 10]$, $\varphi = 30$, $g = 20$, $\phi_1 = 17$ and $\phi_2 = 25$, and $L_{att}^{largest} = 6$. We save the lossless image in png format similar to the experimental image.

As shown in Fig. 6, our algorithm can achieve progressively clear image restoration. In addition, our test results for

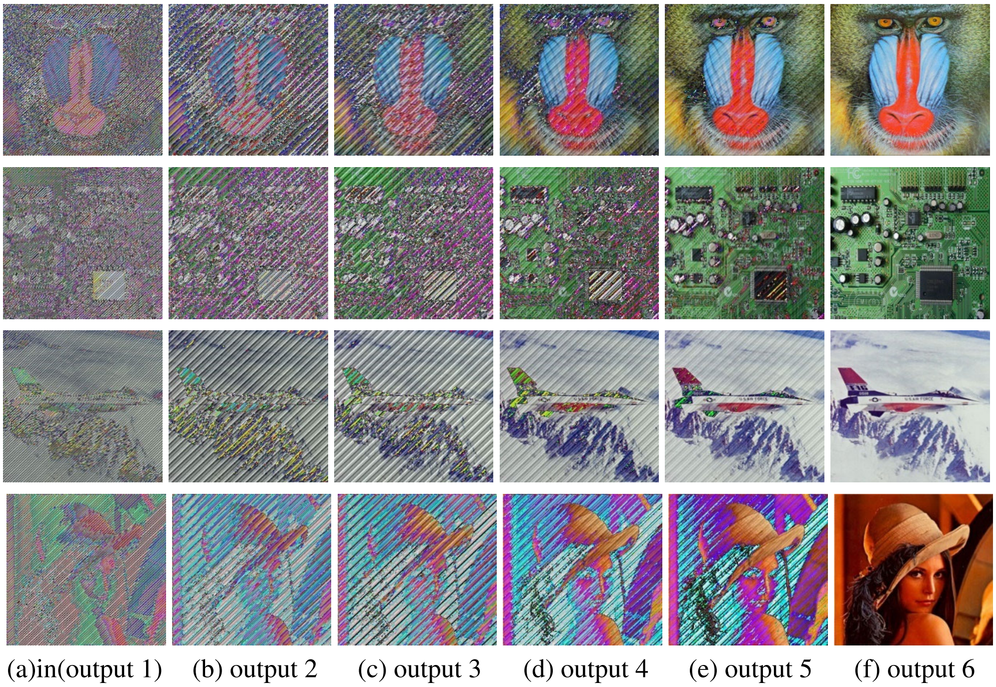


FIGURE 6. The result images for lossless progressive restoration.

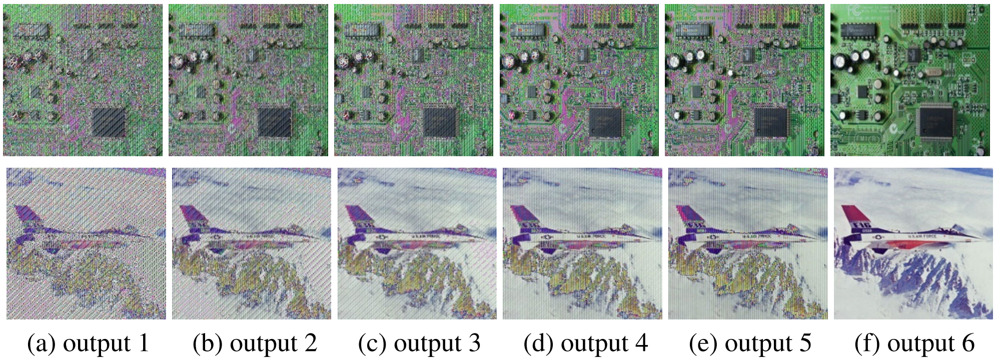


FIGURE 7. The images resulting from RCT+progressively clear restoration.

the PSNR show that the restored images are all identical to the original images.

3) REVERSIBLE CONVOLUTION TRANSFORMATION + PROGRESSIVELY CLEAR IMAGE RESTORATION

This experiment adds an inverse convolution transformation to the previous experiment, so each case needs to display 7 images, which increases the need for page space. To save space, we only display six restored images in Fig. 7 with "output i", and make them correspond to the six privilege levels of the user. Here, the input image is more blurred than the image of "output 1".

As shown in Column 5 (output 5) of Fig. 7, under the condition of lossless full restoration, some noise still remains in the image. To completely remove this noise (as shown in Column 6), an inverse convolution transformation must be performed. Therefore, our method that protects the security of image information using noise in multiple ways has a better protection effect on image information.

4) EFFECTS OF SOME CLASSIC ATTACKING ALGORITHMS ON THE PROTECTED IMAGES

For the progressively clear image restoration algorithms used above, it is natural to think of their anti-attack capabilities. To this end, we use two classic dehazing algorithms and two classic denoising algorithms to test the anti-attack capabilities of the algorithms provided in this paper. These classic algorithms are the multi-scale Retinex with color restoration (MSRCR) algorithm, the dehazing algorithm based on the dark channel prior, the Wiener filtering algorithm, and the inverse filtering algorithm. Below, we take the level 5 ("output 5" in Figs. 4, 6, and 7) circuit image and aircraft

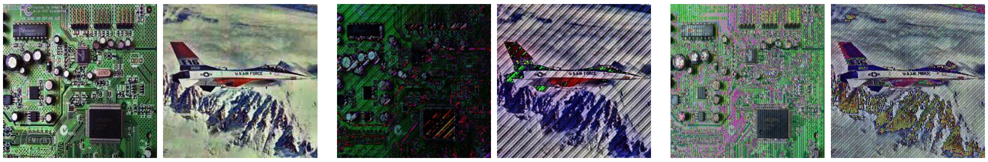

(a) result images of Fig. 4 (b) result images of Fig. 6 (c) result images of Fig. 7

FIGURE 8. The image restoration results based on MSRCR.

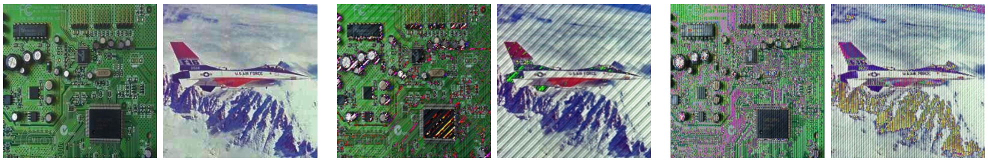

(a) result images of Fig. 4 (b) result images of Fig. 6 (c) result images of Fig. 7

FIGURE 9. The image restoration results based on the dark channel prior.

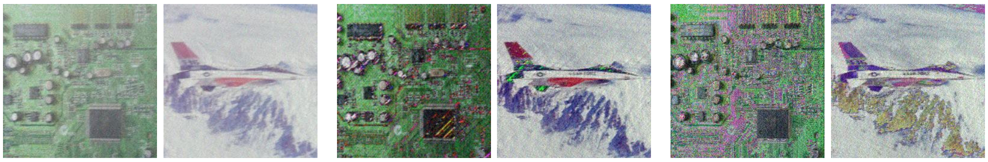

(a) result images of Fig. 4 (b) result images of Fig. 6 (c) result images of Fig. 7

FIGURE 10. The image restoration results based on Wiener filtering.

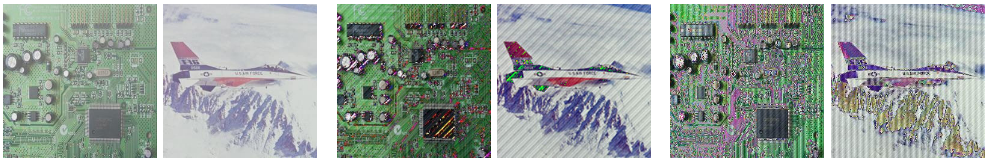

(a) result images of Fig. 4 (b) result images of Fig. 6 (c) result images of Fig. 7

FIGURE 11. The image restoration results based on inverse filtering.

image as the input images to show the restoration ability of these four classic algorithms, as shown in Figs. 8-11, where (a), (b), and (c) are the restoration results of the input images, respectively.

As shown in Figs. 8 and 9, the dehazing algorithm based on MSRCR and the dark channel prior cannot restore the hazed image to the original image. According to our test results, their highest PSNRs of each channel of the dehazed image are 25.18*dB* and 25.28*dB*, respectively.

To investigate the anti-attack capability of our algorithms, we further tested the dehazing intensity of these two algorithms. Based on the 0.05-level haze density we use in the experiments, from a visual point of view, the MSRCR-based algorithm can remove approximately 1/2 layer of haze, and the algorithm based on the dark channel prior can remove approximately two layers of haze. Therefore, by thickening the density of each level of haze, our loss image restoration strategy can effectively resist the dehazing attacks of these two dehazing algorithms. Further experimental results confirm our analysis, which will not be shown here.

TABLE 2. Comparison of PSNRs with the existing literature.

Study	Restoration scheme	PSNR(dB)
[6]	Hybrid graph Laplacian regularization	30.28
[7]	Based on generative adversarial network	33.92
[8]	Based on Laplacian pyramid adversarial network	23.70
[9]	On-demand learning under convolutional neural network	31.32
[10]	Deterministic annealing	34.92
[42]	An unequal power allocation method (4 layers)	34.29
[43]	Progressive transmission of JPEG2000 (30 layers)	36.76
[44]	Progressive transmission of JPEG2000 + link adaption strategy for system parameters (4 layers)	37.61
Ours	Based on the inverse process of atmospheric scattering	49.88

In addition, as shown by the experimental results in Figs. 8 and 9, the dehazing algorithms based on MSRCR and the dark channel prior are basically ineffective for the 5th level (output 5) images generated by our other two algorithms.

TABLE 3. A comparison with the existing literature with image display.

Scheme	Access structure	Length of private key	Size of ciphertext	Number of bilinear pairings	Multiple centers
Yuan's [2]	tree	$(2n+1)G_1$	$(2n+1)G_1+1G_2$	2n	No
Picazo-Sanchez's [3]	LSSS	$(2n)G_1$	$3lG_1+(l+1)G_2$	$4l$	Yes
Ma's [4]	tree	$(2n+1)G_1$	$(2n+1)G_1+1G_2$	2n	No
Ours	tree	$(2n+2)G_1$	$1G_1+1G_2$	1	Yes

Note: n represents the number of attributes selected by the user; l refers to the number of rows of the policy matrix $(\mathcal{M}, \rho)$. In addition, since the multi-authority CP-ABE in [45] is used in [3], when we compare it with the algorithm in [3], the real comparison work is the multi-authority CP-ABE in [45].

We also see that Wiener filtering and inverse filtering do not produce an effective denoising effect regardless of which algorithm we use.

It should be noted that, in order to distinguish images with different sharpness, this paper presents only six restoration results at different sharpness levels. We also perform experiments with image restoration at other sharpness levels. From the experimental results, the lower the sharpness level of the input image is, the higher the computational cost when restoring to the state of complete clarity. Nevertheless, due to the different levels of privileges of different users, sometimes it is not necessary to restore to a completely clear state for protecting images. For example, taking the parameters of Fig. 4 as an example, if 100 sharpness levels are set (the lowest level is 1), then when the image with level 1 of sharpness is input, the image is still blurred enough when it is restored to the 50^{th} level; when it is restored to the 90^{th} level, it has only a weak display effect; and when it is restored to the 96^{th} level, it is roughly equivalent to the 3^{rd} level (output 3) in Fig. 4. Obviously, the more levels of sharpness, the better it is to protect image information security. Moreover, the greater the difference between the two different levels of sharpness, the more resistant the protected image is to various types of image restoring attacks.

C. COMPARISON WITH THE EXISTING ALGORITHMS

In this section, we compare our algorithms with other ones in two aspects, namely, PSNR and CP-ABE performance. Among them, the comparisons on PSNR are aimed at the performance of existing progressive restoration algorithms, while for CP-ABE, we compare the performance of the CP-ABE algorithm proposed in this paper with other algorithms. We first show the comparison results for PSNRs.

1) COMPARISON ON PSNR

Only the PSNR value of the last restored image is measured here.

In Table 2, we mainly compare it with the progressive image restoration algorithm with loss. The progressive restoration technologies are used in [6]–[10], and we also compare our algorithm with the methods using progressive transmission technology [42]–[44] for completeness. As shown in Table 2, in the case of loss restoration, the PSNR that we provide is at least 32.62% higher than those provided by the comparison objects.

2) PERFORMANCE COMPARISON FOR THE IMPROVED CP-ABE ALGORITHMS

Here, we divide the existing literature to be compared into two categories, studies with image display and studies without image display because the ideas in the literature with image display are basically consistent with ours. The reason that we compare our algorithm with the literature without image display is to show the good performance of our CP-ABE algorithm in terms of the ciphertext length, the number of bilinear pairing calculations, and the flexibility of the access policy. In Table 3, we provide a comparison with the existing literature with image display. The comparison results for the literature without image display are shown in Table 4.

As seen in Table 3, our solution can implement a flexible access policy and has constant-size ciphertext and constant bilinear pairing calculations; thus, the performance of our solution is better than that of the others in these aspects. For those people who upload images to a social platform using smart devices anytime and anywhere, this level of protection would satisfy their demands very well.

As shown in Table 4, to make the comparison results more comprehensive, we select these access structures: tree type, LSSS type, AND-gate type, AND, NOT-gate type, and extended LSSS type [31], [46]; we then add (t, s)-threshold type [47]. These structures are all commonly used at present. Among these structures, the tree structure, especially that in [1], can support AND, OR, NOT, and threshold operations. Although we can convert the tree-type structure to the structure of the LSSS type, based on the results of [19] and our knowledge, the ciphertext generated by such access structures is not a constant size.

In summary, our algorithm is superior to the algorithms without image restoration in the performance of CP-ABE. For example, our ciphertext length is the smallest. For the algorithms with bilinear pairing calculations, we have the fewest bilinear pairings. At the same time, our algorithm can support AND, OR, NOT, and threshold operation. Although the CP-ABE algorithm in [27] constructed by *RSA* has no bilinear pairing calculation, the length of its constant-size ciphertext is not short, and it is debatable whether the CP-ABE without the bilinear pairing calculation is consistent with the original design intention of CP-ABE.

TABLE 4. A comparison with the existing literature without image display.

Scheme	Access structure	Length of private key	Size of ciphertext	Number of bilinear pairings	Multiple centers
BSW [1]	tree	$(2n+1)G_1$	$(2n+1)G_1$	2n	No
Yang's [17]	LSSS	nG_1	$2lG_1+(l+1)G_2$	$2l$	Yes
Han's [25]	AND,NOT-gate	$(n+1)G_1$	$3G_1+1G_2$	2	No
Odelu's [27]	AND-gate	$2G_1^{'}$	$3G_1^{'}+L$	0	No
Xiong's [28]	LSSS	$(5n+2)G_1$	$(6l+3)G_1$	3	No
Han's [31]	LSSS	$(n+6)G_1$	$(2l+3)n^{'}G_1+1G_2$	$(2l+4)n^{'}$	Yes
Ohtake's [46]	LSSS	$(n+2)G_1$	$(2l+1)G_1+1G_2$	$2l+1$	No
Susilo's [47]	(t,s)-threshold	$(n+m)G_1$	$2G_1+1G_2$	2	No
Ours	tree	$(2n+2)G_1$	$1G_1+1G_2$	1	Yes

Note: n represents the number of attributes selected by the user; l refers to the number of rows of the policy matrix $(\mathcal{M},\rho)$. L represents the length of the plaintext; $n^{'}$ denotes the number of selected AC; m refers to the maximum number of attributes to be encrypted.

IX. CONCLUSION AND FUTURE WORK

With the increasing in social software use, there are more cases of people uploading images to social platforms at any time. To effectively control the degree to which users can acquire image information, based on the characteristics of CP-ABE, we propose a strategy for restoring an image corresponding to a level of sharpness based on the accessor's attribute level. In our strategy, we improve the original CP-ABE in [1] and provide methods that progressively restore the protected image for the user. Our improved CP-ABE not only supports AND-gate, OR-gate, NOT-gate, and threshold operations but also allows the data accessors to store shorter ciphertexts and offers a constant number of bilinear pairings that need to be calculated when decrypting. In addition, we offer corresponding policy updating algorithms for attribute revocation, prove the correctness and security of our proposed CP-ABE, and provide the algorithms for progressively and clearly restoring images. Finally, we propose a progressively clear image restoration strategy based on our proposed CP-ABE method and the algorithms for restoring images. Both the experimental results and the theoretical analysis demonstrate the superior performance of our algorithms in terms of progressively clear image restoration and attribute-based access control for protecting image information security.

In the future, we will develop a system that can be run on smart terminals such as mobile phones, and conduct research on the ABE in heterogeneous environments. In addition, we will study the common attribute revocation and the accompanying attribute revocation. Moreover, we will also study a lossless progressively clear image restoration scheme based on the algorithm proposed in this paper that can make the images of different sharpness have a finer and smoother display.

REFERENCES

[1] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy*, May 2007, pp. 321–334.

[2] L. Yuan, D. McNally, A. Küpçü, and T. Ebrahimi, "Privacy-preserving photo sharing based on a public key infrastructure," *Proc. SPIE*, vol. 9599, Sep. 2015, Art. no. 95991I. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/9599/95991I/Privacy-preserving-photo-sharing-based-on-a-public-key-infrastructure/10.1117/12.2190458.short

[3] P. Picazo-Sanchez, R. Pardo, and G. Schneider, "Secure photo sharing in social networks," in *Proc. IFIP Int. Conf. ICT Syst. Secur. Privacy Protection*, vol. 502. Cham, Switzerland: Springer, 2017, pp. 79–92.

[4] C. Ma and C. W. Chen, "Secure media sharing in the cloud: Two-dimensional-scalable access control and comprehensive key management," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2014, pp. 1–6.

[5] H. Sun, H. Luo, T.-Y. Wu, and M. S. Obaidat, "Adaptive and safe presentation strategy of image information on social platform," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–7.

[6] D. Zhai, X. Liu, D. Zhao, H. Chang, and W. Gao, "Progressive image restoration through hybrid graph Laplacian regularization," in *Proc. Data Compress. Conf.*, Mar. 2013, pp. 103–112.

[7] Y. Wang, F. Perazzi, B. McWilliams, A. Sorkine-Hornung, O. Sorkine-Hornung, and C. Schroers, "A fully progressive approach to single-image super-resolution," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2018, pp. 864–873.

[8] Q. Wang, H. Fan, G. Sun, Y. Cong, and Y. Tang, "Laplacian pyramid adversarial network for face completion," *Pattern Recognit.*, vol. 88, pp. 493–505, Apr. 2019.

[9] R. Gao and K. Grauman, "On-demand learning for deep image restoration," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2017, pp. 1086–1095.

[10] C. Knaus and M. Zwicker, "Progressive image denoising," *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 3114–3125, Jul. 2014.

[11] X. Yan, Y. Lu, and L. Liu, "A general progressive secret image sharing construction method," *Signal Process., Image Commun.*, vol. 71, pp. 66–75, Feb. 2019.

[12] H.-C. Chao and T.-Y. Fan, "Random-grid based progressive visual secret sharing scheme with adaptive priority," *Digit. Signal Process.*, vol. 68, pp. 69–80, Sep. 2017.

[13] Y.-X. Liu, C.-N. Yang, C.-M. Wu, Q.-D. Sun, and W. Bi, "Threshold changeable secret image sharing scheme based on interpolation polynomial," *Multimedia Tools Appl.*, vol. 78, no. 13, pp. 18653–18667, 2019.

[14] Z. Qian, X. Zhang, and G. Feng, "Reversible data hiding in encrypted images based on progressive recovery," *IEEE Signal Process. Lett.*, vol. 23, no. 11, pp. 1672–1676, Nov. 2016.

[15] Y. Cheng, H. Zhou, J. Ma, and Z. Wang, "Efficient CP-ABE with non-monotonic access structures," in *Proc. Int. Conf. Cloud Comput. Secur.* Cham, Switzerland: Springer, 2017, pp. 315–325.

[16] C.-J. Wang and J.-F. Luo, "A key-policy attribute-based encryption scheme with constant size ciphertext," in *Proc. 8th Int. Conf. Comput. Intell. Secur.*, Nov. 2012, pp. 447–451.

[17] K. Yang, X. Jia, K. Ren, R. Xie, and L. Huang, "Enabling efficient access control with dynamic policy updating for big data in the cloud," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 2013–2021.

[18] X. Dong, J. Yu, Y. Luo, Y. Chen, G. Xue, and M. Li, "Achieving an effective, scalable and privacy-preserving data sharing service in cloud computing," *Comput. Secur.*, vol. 42, pp. 151–164, May 2014.

[19] S. Canard and V. C. Trinh, "Private ciphertext-policy attribute-based encryption schemes with constant-size ciphertext supporting CNF access policy," *IACR Cryptol. ePrint Arch.*, vol. 2015, p. 891, 2015. [Online]. Available: https://dblp.uni-trier.de/rec/bibtex/journals/iacr/CanardT15

[20] J. Herranz, F. Laguillaumie, and C. Ràfols, "Constant size ciphertexts in threshold attribute-based encryption," in *Proc. Int. Workshop Public Key Cryptogr.* Berlin, Germany: Springer, 2010, pp. 19–34.

[21] H. Lin, A. Cao, X. Liang, and J. Shao, "Secure threshold multi authority attribute based encryption without a central authority," in *Proc. Int. Conf. Cryptol. India.* Berlin, Germany: Springer, 2008, pp. 426–436.

[22] N. Attrapadung, J. Herranz, F. Laguillaumie, B. Libert, E. de Panafieu, and C. Ràfols, "Attribute-based encryption schemes with constant-size ciphertexts," *Theor. Comput. Sci.*, vol. 422, pp. 15–38, Mar. 2012.

[23] J. Li, X. Chen, S. S. M. Chow, Q. Huang, D. S. Wong, and Z. Liu, "Multi-authority fine-grained access control with accountability and its application in cloud," *J. Netw. Comput. Appl.*, vol. 112, pp. 89–96, Jun. 2018.

[24] Z. Zhang, C. Li, B. B. Gupta, and D. Niu, "Efficient compressed ciphertext length scheme using multi-authority CP-ABE for hierarchical attributes," *IEEE Access*, vol. 6, pp. 38273–38284, 2018.

[25] J. Han, Y. Yang, J. K. Liu, J. Li, K. Liang, and J. Shen, "Expressive attribute-based keyword search with constant-size ciphertext," *Soft Comput.*, vol. 22, no. 15, pp. 5163–5177, Aug. 2017.

[26] X. Li, S. Tang, L. Xu, H. Wang, and J. Chen, "Two-factor data access control with efficient revocation for multi-authority cloud storage systems," *IEEE Access*, vol. 5, pp. 393–405, 2016.

[27] V. Odelu, A. K. Das, M. K. Khan, K. R. Choo, and M. Jo, "Expressive CP-ABE scheme for mobile devices in IoT satisfying constant-size keys and ciphertexts," *IEEE Access*, vol. 5, pp. 3273–3283, 2017.

[28] 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.

[29] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, 2007, pp. 195–203.

[30] L. Li, T. Gu, L. Chang, Z. Xu, Y. Liu, and J. Qian, "A ciphertext-policy attribute-based encryption based on an ordered binary decision diagram," *IEEE Access*, vol. 5, pp. 1137–1145, 2017.

[31] J. Han, W. Susilo, Y. Mu, J. Zhou, and M. H. Au, "PPDCP-ABE: Privacy-preserving decentralized ciphertext-policy attribute-based encryption," in *Proc. Eur. Symp. Res. Comput. Secur.* Cham, Switzerland: Springer, 2014, pp. 73–90.

[32] E. J. McCartney, *Optics of the Atmosphere: Scattering by Molecules and Particles*, vol. 421. New York, NY, USA: Wiley, 1976, pp. 76–77.

[33] A. Wang, W. Wang, J. Liu, and N. Gu, "Aipnet: Image-to-image single image dehazing with atmospheric illumination prior," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 381–393, Jan. 2019.

[34] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[35] W. Bae, J. Yoo, and J. C. Ye, "Beyond deep residual learning for image restoration: Persistent homology-guided manifold simplification," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2017, pp. 145–153.

[36] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 6721–6729.

[37] S. R. M. Penedo, F. A. M. Cipparrone, and J. F. Justo, "Digital image inpainting by estimating wavelet coefficient decays from regularity property and Besov spaces," *IEEE Access*, vol. 7, pp. 3459–3471, 2019.

[38] M. Isogawa, D. Mikami, D. Iwai, H. Kimata, and K. Sato, "Mask optimization for image inpainting," *IEEE Access*, vol. 6, pp. 69728–69741, 2018.

[39] J. Li, G. Li, and H. Fan, "Image dehazing using residual-based deep CNN," *IEEE Access*, vol. 6, pp. 26831–26842, 2018.

[40] J. Li, N. Chen, and Y. Zhang, "Extended file hierarchy access control scheme with attribute based encryption in cloud computing," *IEEE Trans. Emerg. Topics Comput.*, to be published.

[41] Y. Zhang, J. Li, and H. Yan, "Constant size ciphertext distributed CP-ABE scheme with privacy protection and fully hiding access structure," *IEEE Access*, vol. 7, pp. 47982–47990, 2019.

[42] G. Javadi, A. Hajshirmohammadi, and J. Liang, "Power and sub-channel optimization of JPEG 2000 image transmission over OFDM-based cognitive radio networks," *Signal Process., Image Commun.*, vol. 58, pp. 157–164, Oct. 2017.

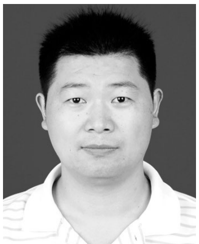
[43] C. Y. Bi and J. Liang, "Joint source-channel coding of jpeg 2000 image transmission over two-way multi-relay networks," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3594–3608, Jul. 2017.

[44] M. Mhamdi, C. Perrine, A. Zribi, Y. Pousset, C. Olivier, and A. Bouallègue, "Soft decoding algorithms for optimized JPEG 2000 wireless transmission over realistic MIMO-OFDM systems," *Signal Process., Image Commun.*, vol. 52, pp. 41–53, Mar. 2017.

[45] Y. Rouselakis and B. Waters, "Efficient statically-secure large-universe multi-authority attribute-based encryption," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2015, pp. 315–332.

[46] G. Ohtake, R. Safavi-Naini, and L. F. Zhang, "Outsourcing scheme of ABE encryption secure against malicious adversary," *Comput. Secur.*, vol. 86, pp. 437–452, Sep. 2019.

[47] W. Susilo, G. Yang, F. Guo, and Q. Huang, "Constant-size ciphertexts in threshold attribute-based encryption without dummy attributes," *Inf. Sci.*, vol. 429, pp. 349–360, Mar. 2018.



HUAIBO SUN received the M.S. degree in applied mathematics from The PLA Information Engineering University, China, in 2006. He is currently pursuing the Ph.D. degree with the Internet of Things Laboratory, Beijing University of Posts and Telecommunications, Beijing. His research interests include the Internet of Things, data hiding, information security, image processing, and crowd computing.

HONG LUO received the B.S., M.S., and Ph.D. degrees from the Beijing University of Posts and Telecommunications, Beijing, China, in 1990, 1993, and 2006, respectively. From 2004 to 2005, she held a visiting position at the Department of Computer Science and Engineering, The University of Texas at Arlington. She was a Visiting Professor with the Helsinki University of Technology, in 2008. She is currently a Professor with the School of Computer Science, Beijing University of Posts and Telecommunications. She is also a Research Member of the Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia. Her research interests include the Internet of Things, wireless networking, sensor networks, smart environments, and communication software. She was a recipient of the New Century Excellent Talents from the University of China, in 2008.

YAN SUN received the B.S. degree from Beijing Jiaotong University, in 1992, and the M.S. and Ph.D. degrees from the Beijing University of Posts and Telecommunications, in 1996 and 2007, respectively. She is currently a Professor with the School of Computer Science, Beijing University of Posts and Telecommunications, China. She is also a Research Member of the Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia. Her research interests include the Internet of Things, sensor networks, smart environments, and embedded systems.