IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 19, 2021, accepted March 10, 2021, date of publication March 17, 2021, date of current version March 23, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3066212

Efficient Revocable Multi-Authority Attribute-Based Encryption for Cloud Storage

YANG MING, (Member, IEEE), BAOKANG HE, AND CHENHAO WANG
School of Information Engineering, Chang'an University, Xi'an 710064, China

Corresponding author: Yang Ming (yangming@chd.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 62072054.

ABSTRACT As is known, attribute-based encryption (ABE) is usually adopted for cloud storage, both for its achievement of fine-grained access control over data, and for its guarantee of data confidentiality. Nevertheless, single-authority attribute-based encryption (SA-ABE) has its obvious drawback in that only one attribute authority can assign the users' attributes, enabling the data to be shared only within the management domain of the attribute authority, while rendering multiple attribute authorities unable to share the data. On the other hand, multi-authority attribute-based encryption (MA-ABE) has its advantages over SA-ABE. It can not only satisfy the need for the fine-grained access control and confidentiality of data, but also make the data shared among different multiple attribute authorities. However, existing MA-ABE schemes are unsuitable for the devices with resources-constraint, because these schemes are all based on expensive bilinear pairing. Moreover, the major challenge of MA-ABE scheme is attribute revocation. So far, many solutions in this respect are not efficient enough. In this paper, on the basis of the elliptic curves cryptography, we propose an efficient revocable multi-authority attribute-based encryption (RMA-ABE) scheme for cloud storage. The security analysis indicates that the proposed scheme satisfies indistinguishable under adaptive chosen plaintext attack assuming hardness of the decisional Diffie-Hellman problem. Compared with the other schemes, the proposed scheme gets its advantages in that it is more economical in computation and storage.

INDEX TERMS Attribute-based encryption, multi-authority, revocation, elliptic curve cryptography, cloud storage.

I. INTRODUCTION

Cloud storage is an application pattern of cloud computing [1] to store massive data, so more and more individuals and organizations shift their data from local computers to cloud. However, this new paradigm poses a serious threat to the privacy of their owners, since the data might be accessed and analyzed by the cloud server providers for illegal or monetary purposes.

To solve this problem, people have figured out a variety of approaches. One common way is to resort to the traditional public key encryption technology to encrypt data, but the data owners fail to have fine-grained access to their data flexibly. Accordingly, Sahai and Waters [2] advanced a new way of encryption, attribute-based encryption (ABE). It used to be considered one of the most promising technologies for cloud storage, since it ensures the data owners to enjoy non-interactive and fine-grained control over encrypted data.

Since then, many single-authority attribute-based encryption (SA-ABE) schemes [2]–[9] have been put forward. In these schemes, it is required that only one trusted attribute authority administers the attributes and distributes the corresponding secret keys of attributes to the data consumers. This mechanism may not meet the practical requirements in cloud storage, when data consumers' attributes are distributed by multiple different attribute authorities. For example, when a data owner intends to share the data with a targeted data consumer holding the attribute "Professor" from a university and the attribute "Engineer" from a research institution, obviously SA-ABE scheme cannot be applied to this scenario. To deal with this problem, many researchers [10]–[24] turn to multi-authority attribute-based encryption (MA-ABE), so that secret keys of attributes are issued to data consumers with the corresponding privileges for different attribute authorities respectively. There exists

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

two kinds of multi-authority ABE schemes, namely centralized multi-authority ABE and decentralized multi-authority ABE, the difference between them is whether the key is distributed by center authority. When the key is distributed by central authority, we can consider it as centralized multi-authority ABE scheme [10]. When the key is distributed by attribute authority, we can consider it as decentralized multi-authority ABE scheme [2]–[9], [11]–[23], [25], [26].

From the perspective of practical application, the following challenges should be solved before applying MA-ABE in cloud storage system. One of the major challenges is the highly computational overhead, since the existing MA-ABE schemes [10]–[23] are all based on the expensive bilinear pairing operations, hinders the further development of MA-ABE schemes on the resource-constrained devices. The other challenge is the attribute revocable, since multiple data consumers may share the same attribute, and each data consumers may possess multiple different attributes, result in that revocation for anyone attribute may influence the other data consumers in the cloud storage system. Although re-encrypting the data is a method [19] to solve this problem, it will generate high computation cost. Another technology [24] is to introduce a timestamp into every attribute, but it is not achieved immediate revocation.

This paper involves the construction of an efficient RMA-ABE scheme for cloud storage. Our main contributes are as follows:

- First, based on the elliptic curve cryptography (ECC), an efficient RMA-ABE scheme is proposed for cloud storage, so that bilinear pairing operations will be no longer needed. In the proposed scheme, we use the linear secret sharing schemes (LSSS) to boost the expressiveness of access policy and add version key to attribute to realize immediate attribute revocation.
- Second, the security analysis indicates that under the decisional Diffie-Hellman (DDH) assumption, the proposed RMA-ABE scheme achieves the indistinguishability against the choose plaintext attack (IND-CPA), and satisfies collision resistant and forward secrecy.
- Finally, the performance evaluation of the scheme indicates lower the computation cost and lower storage overhead than other schemes.

The rest of this paper is organized as follows. Section II introduces the related work. Preliminaries are given in Section III. The concrete RMA-ABE scheme is described in Section IV. In Sections V and VI, the security and performance analysis of the proposed scheme is shown, respectively. In Section VII, this paper is concluded.

II. RELATED WORK

The ABE scheme is introduced by Sahai and Waters [2]. With ABE, data owner can share his/her data encrypted with the targeted data consumers, with no knowledge of their public keys or identities, ensuring ABE schemes to achieve fine-grain and flexible access control in cloud storage. The notion of key-policy attribute-based encryption (KP-ABE) was put forward by Goyal *et al*. [3], that the data consumers' secret keys are relevant to access structures, and the ciphertext is related to certain attributes. Then, Bethencourt and Sahai [4] introduced the concept of ciphertext-policy attribute-based encryption (CP-ABE), that data consumers' secret keys are relevant to some attributes, and ciphertexts are relevant to access structures. The CP-ABE scheme is more applicable to access control, since data owners can determine access structures. Thus many scholars pay attention to CP-ABE schemes [5]–[9].

In ABE schemes [2]–[9], there is only one attribute authority administering all attributes. However, problem arises as attributes are distributed by multiple attribute authorities, so Chase [10] proposed MA-ABE scheme. According to Chase's scheme several different attribute authorities issue attributes and secret keys of attributes to data consumers. Nevertheless, the privacy of data owner is not been protected because the central authority can decrypt all the ciphertexts. Then, Chase and Chow [11] introduced the MA-ABE scheme without the trusted authority, but the data consumers need at least one attribute from every attribute authority, so this scheme is not practical enough. Lewko and Waters [12] devised a CP-ABE scheme with fully decentralized multi-authority, needing no cooperation among the attribute authorities, so from every attribute authority, data consumers can possess any number of attributes.

The majority construction of the above MA-ABE [10]–[12] schemes involves bilinear pairing operations scaling with the attribute number, resulting in exorbitant computational overhead in both encryption and decryption phases.

To lower the computation overhead at encryption phase, Zhang *et al*. [13] offered an MA-ABE scheme in mobile cloud by using the technique of offline/online encryption. According to this scheme, all major encryption computation can be shifted to the offline phase, so the data owners only perform a few online computations. To reduce the computation cost at decryption phase, data consumers can outsource the decryption computing to the third-party servers. Yang *et al*. [14] presented a supporting decryption outsourcing multi-authority CP-ABE scheme, where data consumers do not need to perform any complex bilinear pairing operations in decryption phase. To guarantee the security of data, data consumers must be able to examine the correctness and completeness of the received ciphertext partially decrypted. Then, Li *et al*. [15] proposed a securely outsourcing MA-ABE scheme with checkability, without the need of a trusted central authority, and it can check the correctness and completeness of outsourcing decryption. Xu *et al*. [16] put forward a decentralized ABE scheme for cloud computing. It can provide both online/offline encryption and outsourcing decryption, and its security is guaranteed in the random oracle model (ROM). Belguith *et al*. [17] introduced the outsourcing MA-ABE scheme for cloud assisted IoT, in which users' privacy can be protected by hiding access policy. Sethi *et al*. [18] constructed the MA-ABE scheme,

supporting both white-box traceability and policy updating, as well as outsourcing decryption over large attribute universe. Although the MA-ABE schemes [14]–[18] are effective by reducing encryption or decryption computational overhead, these schemes do not support attribute revocation.

For MA-ABE systems, when data consumer's attributes are changed, the access permission for the data consumer also ought to be altered timely and effectively. Therefore, efficient attribute revocation is necessary for MA-ABE schemes. Ruj *et al*. [19] introduced the MA-ABE scheme able to support revocation of attributes. The attribute revocation method [19] results in high costs in computation and communication, since once an attribute is revoked, a data owner is supposed to re-encrypt the data before sending the updated ciphertext to all non-revoked data owner. Accordingly, Yang and Jia [20] presented a MA-ABE scheme able to revoke attributes with efficiency. They introduced a version key into each attribute, so the moment an attribute revocation occurs, the revoked attribute's version key will be changed, and the attribute authority generates updated keys and transmits them to each non-revoked data consumers according to the new version keys, so that the data consumers can update their own secret keys. Li *et al*. [21] figured out a new approach to revoke attributes. In this approach, a set of attribute group keys are used to re-encrypt ciphertexts by a third-party sever. The scheme also supports decryption outsourcing, so data consumers can enjoy great reduction of the decryption overhead. Basing on the revocation method [21], Liu *et al*. [22] formulated a more practical MA-ABE scheme, combining the attribute revocation, outsourcing decryption, and policy updating. Fan *et al*. [23] presented an attribute revocation MA-ABE scheme for privacy computing through modifying the version number of revoked attributes, and realized encryption and decryption computations outsourcing.

Ding *et al*. [25] proposed a novel efficient pairing-free CP-ABE based on elliptic curve cryptography, which is pairing-free and able to revoke attribute easily, but the attribute revocation method [25] has disadvantages. When a user's attribute is revoked, other user with this attribute will be affected. Wang *et al*. [26] presented efficient and secure CP-ABE without pairing, which can achieve attribute revocation. Although [25], [26] are pairing-free based on ABE, they only support single authority, not multi-authority.

III. PRELIMINARIES

A. NOTATIONS

In order to promote the understanding, Table 1 presents the notations applied throughout the paper.

B. ACCESS STRUCTURE

$\{P_1, P_2, \cdots, P_n\}$ is denoted as an attribute universe. A collection $\mathbb{A} \subseteq 2^{\{P_1,P_2,\cdots,P_n\}}$ is monotone on $\forall B, C$: if $B \in \mathbb{A}, B \subseteq C$, then $C \in \mathbb{A}$. A collection $\mathbb{A}$ of non-empty subsets of $\{P_1, P_2, \cdots, P_n\}$ is an access structure, such as $\mathbb{A} \subseteq 2^{\{P_1,P_2,\cdots,P_n\}}/\{\emptyset\}$. The set in $\mathbb{A}$ is designated as the authorized set, and the set outside $\mathbb{A}$ as the unauthorized set.

TABLE 1. Notations.

Symbol	Definition
uid	The identity of data consumer
aid_k	The identity of k-th attribute authority
AA_{aid}	The attribute authority with index aid
x_{aid}	The attribute of AA_{aid}
$VK_{x_{aid}}$	The version key of attribute x_{aid}
$PK_{x_{aid}}$	The public key of attribute x_{aid}
$UK_{x_{aid}}$	The update key of attribute x_{aid}
S_{aid}	The attribute set governed by AA_{aid}
S_{uid}	The attribute set of uid
$S_{uid,aid}$	The attribute set of uid distributed by AA_{aid}
APK_{aid}	The public key of AA_{aid}
ASK_{aid}	The secret key of AA_{aid}
$SK_{uid,aid}$	The secret key of uid distributed by AA_{aid}
U_A	The set of attribute authorities
$LSSS$	Linear secret sharing schemes
(M, ρ)	Access structure expressed by LSSS
I_A	The set of attribute authorities related to (M, ρ)
n_A	The number of attribute authorities in I_A
S_R	The set of revoked attributes
n_R	The number of revoked attributes in S_R

C. LINEAR SECRET SHARING SCHEMES

Let $\mathbb{P}$ be an attribute universe and q be a prime. Call a secret sharing scheme Π over $\mathbb{P}$ linear secret sharing schemes (LSSS) if:

(1) For each attribute, the shares of a secret $s \in \mathbb{Z}_p$ form a vector over $\mathbb{Z}_p$.

(2) For each access structure T on $\mathbb{P}$, there is a matrix M known as the share-generating matrix for Π, where the matrix M is l rows and n columns. Let ρ denote a function mapping the row $i \in [1, 2, \cdots, l]$ of M to an attribute. Preset a column vector $\overrightarrow{v} = (s, r_2, \cdots, r_n) \in \mathbb{Z}_q$, in which s represents as the shared secret and $r_2, r_3, \cdots, r_n$ are selected randomly, $M\overrightarrow{v}$ is the vector of l shares of the secret s. The share $(M\overrightarrow{v})_i$ belongs to attribute $\rho(i)$, where $i = [1, l]$.

According to the description of [27], if a LSSS is structured by the definition mentioned above, shared secret s can be reconstructed. The following defines the reconstruction performance: suppose that the access structure T is converted to a LSSS labeling as Π, and defines an authorized set $S \subseteq T$. Set $I = \{i : \rho(i) \in S\}$ represents the rows of Π, where the rows can be mapped to these attributes in S. By choosing a secret value s, the shared values $\{\lambda_i = (M\overrightarrow{v})_i\}_{i\in I}$ are computed. Then, there exists constants $\{c_i \in \mathbb{Z}_q\}_{i\in I}$, able to be used to reconstruct the secret value s by computing $\sum_{i\in I} c_i\lambda_i$.

D. ELLIPTIC CURVE

Millier [28] firstly introduced the elliptic curve cryptography (ECC). $\mathbb{F}_q$ is designated as a prime finite field, where q represents a large prime. Based on $\mathbb{F}_q$ and the equation $y^2 = x^3 + ax + b \bmod q$, the elliptic curve E can be defined, where $\Delta{=}4a^3 + 27b^2 \bmod q \neq 0$ and $a, b \in \mathbb{F}_q$. An additive group $\mathbb{G}$ is made up of the infinite point O and all points on E over $\mathbb{F}_q$. The equation $kP = P + P + \cdots + P(k\ times)$ denotes the operation of scalar multiplication, where P denotes the generator of $\mathbb{G}$.

E. SECURITY ASSUMPTION

DDH Problem: Given an elliptic cyclic group $\mathbb{G}$, its order is a prime q, its generator is P. $a, b \in \mathbb{Z}_q^*$, $R \in \mathbb{G}$ are chosen randomly to form a tuple $(P, aP, bP, R) \in \mathbb{G}$. The task of DDH problem is to distinguish abP from a random element R in $\mathbb{G}$.

The advantage of algorithm $\mathcal{A}$ in solving the DDH problem is as follows:

$$\Pr[\mathcal{A}(P, aP, bP, Z = abP) = 0] - \Pr[\mathcal{A}(P, aP, bP, Z = R) = 0]| \geq \varepsilon$$

DDH Assumption: If no polynomial time algorithms with a non-negligible advantage in solving DDH problem, the DDH assumption holds.

F. SYSTEM MODEL

Figure 1 shows that, the proposed RMA-ABE scheme embraces five entities: central authority (CA), attribute authorities (AAs), cloud service provider (CSP), data owner (DO) and data consumer (DC).

- CA: CA, as central authority, is responsible for initializing the system public parameters, verifying and registering the DCs' and AAs' identities and the CA is not involved in any attribute management and the creation of secret keys that are associated with attributes.
- AAs: AA, an independent attribute authority, which is an independent attribute authority that is responsible for entitling and revoking user's attributes according to their role or identity in its domain. The function of AA is generating public keys and secret keys, and distributing secret keys of attributes to data consumers. In addition, when one or more attributes are revoked, AA operates the update key generation algorithm for non-revoked data consumers.
- CSP: CSP is responsible for data storage. When an attribute is revoked, CSP will provide data access service to data consumers and perform ciphertext update algorithm to help data owners update their ciphertext.
- DO: DO performs such operations as defining the access structure over attributes from one or more AAs, encrypting according to the access structure, and uploading encrypted data to the CSP.
- DC: DC has access to ciphertexts from CSP, and requests their secret keys from corresponding AAs. Only if the DC's attribute set meets the access structure, DC decrypts the ciphertexts successfully with his secret keys.

The proposed RMA-ABE scheme is composed of such five phases as: the system initialization, the secret key generation, the data encryption, the data decryption, and the attribute revocation.

System initialization. This phase consists of four algorithms: *CASetup*, *DCReg*, *AAReg* and *AASetup*.

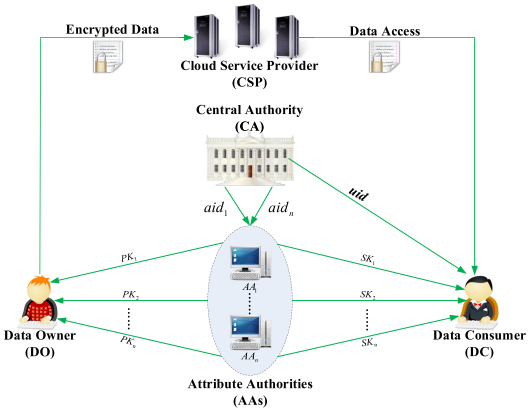

FIGURE 1. System Model.

- $CASetup(\lambda) \rightarrow PP$. This algorithm is implemented by CA. It takes as input a security parameter λ, outputs public system parameters PP.
- $DCReg(info_{DC}) \rightarrow uid$. CA carries out the algorithm, using the DC's information $info_{DC}$ (e.g., name, birthday etc.) as input, and the identity uid as output.
- $AAReg(info_{AA}) \rightarrow aid$. CA performs the algorithm, with AA's information $info_{AA}$ as input, and identity aid as output.
- $AASetup(PP, S_{aid}) \rightarrow APK_{aid}, ASK_{aid}, \{VK_{x_{aid}}, PK_{x_{aid}}\}_{x_{aid} \in S_{aid}}$. AA executes the algorithm, where the system public parameters PP and the attribute set S_{aid} managed by AA_{aid} are used as input, and the outputs are the public and secret key pairs (APK_{aid}, ASK_{aid}) of AA_{aid}, version keys and the public keys of the attributes $\{VK_{x_{aid}}, PK_{x_{aid}}\}_{x_{aid} \in S_{aid}}$ where the attribute set S_{aid} is managed by AA_{aid}.

Secret key generation. This phase is composed of the *SKeyGen* algorithm.

- $SKeyGen(ASK_{aid}, S_{uid,aid}, \{VK_{x_{aid}}\}_{x_{aid} \in S_{uid,aid}}) \rightarrow SK_{uid,aid}$. AA performs the algorithm by entering the secret key ASK_{aid} of AA_{aid}, the DC's attributes set $S_{uid,aid}$ and the corresponding version keys $\{VK_{x_{aid}}\}_{x_{aid} \in S_{uid,aid}}$ to generate secret key $SK_{uid,aid}$ for the DC.

Data encryption. This phase is composed of the *Encrypt* algorithm.

- $Encrypt((M, \rho), \{APK_{aid_k}\}_{aid_k \in I_A}, m) \rightarrow CT$. DO carries out the algorithm by entering access structure (M, ρ), public keys $\{APK_{aid_k}\}_{aid_k \in I_A}$ according to the set I_A related to (M, ρ) and the data m, to output ciphertext CT.

Data decryption. This phase is composed of the *Decrypt* algorithm.

- $Decrypt(CT, \{SK_{uid,aid_k}\}_{aid_k \in I_A}) \rightarrow m$. DC executes the algorithm by entering ciphertext CT, and the secret keys $\{SK_{uid,aid_k}\}_{aid_k \in I_A}$ related to the set I_A, to output the data m.

Attribute revocation. This phase consists of three algorithms: *UKeyGen*, *SKUpdate* and *CTUpdate*.

- $UKeyGen(ASK_{aid'}, \widetilde{x}_{aid'}, VK_{\widetilde{x}_{aid'}}) \rightarrow \widetilde{VK}_{\widetilde{x}_{aid'}}, UK_{\widetilde{x}_{aid'}}$. The algorithm is executed by the corresponding $AA_{aid'}$ administering the revoked attribute $\widetilde{x}_{aid'}$. It takes as input the secret key $ASK_{aid'}$ of $AA_{aid'}$, the version key $VK_{\widetilde{x}_{aid'}}$ of the revoked attribute and the revoked attribute $\widetilde{x}_{aid'}$, outputs the new version key $\widetilde{VK}_{\widetilde{x}_{aid'}}$ and the update key $UK_{\widetilde{x}_{aid'}}$.
- $SKUpdate(SK_{uid,aid'}, UK_{\widetilde{x}_{aid'}}) \rightarrow \widetilde{SK}_{uid,aid'}$. The algorithm is executed by the non-revoked DC. Its inputs are the current secret key $SK_{uid,aid'}$, the update key $UK_{\widetilde{x}_{aid'}}$, and its output is the new secret key $\widetilde{SK}_{uid,aid'}$.
- $CTUpdate(CT, UK_{\widetilde{x}_{aid'}}) \rightarrow \widetilde{CT}$. The algorithm is run by the CSP. Its inputs are the current ciphertext CT, the update key $UK_{\widetilde{x}_{aid'}}$, and its output is the new ciphertext $\widetilde{CT}$.

G. SECURITY MODEL

The security of the RMA-ABE scheme devotes to satisfy the need of confidentiality, namely IND-CPA by defining security game between challenger $\mathcal{C}$ and adversary $\mathcal{A}$.

Initialization. $\mathcal{A}$ specifies an access structure (M^*, ρ^*) to be challenged and a set of corrupted authorities $U'_A \subseteq U_A$, and then transmits them to $\mathcal{C}$.

Setup. $\mathcal{C}$ generates the system public parameters PP by executing the system initialization algorithm, and transmits them to $\mathcal{A}$. If an authority is one of the corrupted authorities in U'_A, $\mathcal{C}$ shall transmit the public keys and secret keys of authority to $\mathcal{A}$. If the authority is a uncorrupted authorities in $U_A - U'_A$, only the public keys of authority need to be transmitted.

Phase 1. The following queries are made by the adversary $\mathcal{A}$:

- *SKey query*. $\mathcal{A}$ makes secret keys queries on $(uid, \{S_{uid,aid_k}\}_{aid_k \in U_A - U'_A})$, where uid stands for a DC's identity, $\{S_{uid,aid_k}\}_{aid_k \in U_A - U'_A}$ is an attributes set belonging to the uncorrupted authorities. $\mathcal{C}$ executes the secret key generation algorithm and returns the corresponding secret keys $\{SK_{uid,aid_k}\}_{aid_k \in U_A - U'_A}$ to $\mathcal{A}$. Note that the access structure (M^*, ρ^*) can't be satisfied by the set $\{S_{uid,aid_k}\}_{aid_k \in U_A - U'_A}$.
- *UKey query*. $\mathcal{A}$ makes update keys queries on S_R, where S_R represents a set of revoked attributes. $\mathcal{C}$ returns the corresponding update keys $\{UK_{x_{aid}}\}_{x_{aid} \in S_R}$ to $\mathcal{A}$ by executing the update key generation algorithm. Note that the set S_R can't meet the access structure (M^*, ρ^*).

Challenge. $\mathcal{A}$ submits two challenge plaintexts m_0^* and m_1^* to $\mathcal{C}$. $\mathcal{C}$ flips randomly a bit $b \in \{0, 1\}$, to encrypt m_b^* according to (M^*, ρ^*) by executing data encryption algorithm. Finally, C returns the challenge ciphertext CT^* to $\mathcal{A}$.

Phase 2. $\mathcal{A}$ could make a certain number of secret keys queries or update keys queries what happens in Phase 1, but it must be restricted: The secret keys or update keys relevant to an attributes set satisfying the access structure (M^*, ρ^*) can't be issued by $\mathcal{A}$.

Guess. After making a guess about b, $\mathcal{A}$ gets a guess b'. If $b' = b$, $\mathcal{A}$ wins the game.

The advantage of the adversary $\mathcal{A}$ is defined as

$$Adv_{\mathcal{A}} = \Pr[b' = b] - 1/2.$$

Definition 1: The RMA-ABE scheme could achieve the indistinguishability under adaptive chosen plaintext attack in the standard model if there is no polynomial time adversary capable of winning in the above game with non-negligible advantage $Adv_{\mathcal{A}}$.

IV. THE PROPOSED SCHEME

This section describes the proposed RMA-ABE scheme in detail.

A. SYSTEM INITIALIZATION

- *CASetup*: CA generates a cyclic group $\mathbb{G}$ with prime order is q, which defines over a finite field $GF(q)$ on the basis of elliptic curve E. Preset P be the generator of cyclic group $\mathbb{G}$. CA selects a hash function $H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$. The system parameters are $PP = \{q, P, \mathbb{G}, H_1\}$.
- *DCReg*: CA chooses a random value $uid \in \mathbb{Z}_q^*$ serving as unique identity of DC.
- *AAReg*: CA chooses a random value $aid \in \mathbb{Z}_q^*$ functioning as unique identity of AA.
- *AASetup*: Each AA_{aid} randomly chooses two numbers $\alpha_{aid}, \beta_{aid} \in \mathbb{Z}_q^*$, and computes $\alpha_{aid}P, \beta_{aid}P$. The secret keys of AA_{aid} are $ASK_{aid} = (\alpha_{aid}, \beta_{aid})$, the public keys of AA_{aid} are $APK_{aid} = (\alpha_{aid}P, \beta_{aid}P)$. For each attribute $x_{aid} \in S_{aid}$, AA_{aid} chooses a random value $\upsilon_{x_{aid}} \in \mathbb{Z}_q^*$ serving as the version key $VK_{x_{aid}} = \upsilon_{x_{aid}}$, to generate the public key $PK_{x_{aid}} = \beta_{aid}\upsilon_{x_{aid}}H_1(x_{aid})P$.

B. SECRET KEY GENERATION

- *SKeyGen*: Before the DC requests secret key from the AA_{aid}, the AA_{aid} needs to authenticate the DC' role or identity. If the DC is certified successful, the AA_{aid} entitles an attributes set $S_{uid,aid}$ to the DC. Then, AA_{aid} computes the DC's secret key as

$$\begin{aligned} SK_{uid,aid} = \{&K_{uid,aid} = \alpha_{aid}/\beta_{aid} + 1/\beta_{aid}, \\ &K_{x_{aid},uid} = \upsilon_{x_{aid}}H_1(x_{aid}) \\ &+H_1(uid)/\beta_{aid}, \forall x_{aid} \in S_{uid,aid}\} \end{aligned}$$

C. DATA ENCRYPTION

- *Encrypt*: To encrypt the data m under an access structure (M, ρ), where M is a matrix with l rows and n columns and ρ is a function, associating each row M_i of M with attribute $\rho(i)$, DO does as follows:
  (1) DO selects two random vectors $\vec{\lambda} = (s, y_2, y_3, \cdots, y_l)$, $\vec{\omega} = (0, z_2, z_3, \cdots, z_l)$ and computes $\lambda_i = M_i \cdot \vec{\lambda}$, $\omega_i = M_i \cdot \vec{\omega}$ for all $i = [1, l]$.
  (2) DO selects a random value $s \in \mathbb{Z}_q^*$ and calculates $C_0 = m + s\sum_{aid_k \in I_A} \alpha_{aid_k}P$, where I_A expresses the set of attribute authorities related to (M, ρ).

(3) DO computes $C_{1,i} = \lambda_i P + \omega_i PK_{\rho(i)}, C_{2,i} = \omega_i \beta_{aid_k} P, C_{3,i} = s\beta_{aid_k} P$ for all $i = [1, l]$.

The ciphertext is $CT = \{C_0, (C_{1,i}, C_{2,i}, C_{3,i})_{i=[1,l]}\}$.

D. DATA DECRYPTION

- *Decrypt*: Suppose that a DC with an attribute set S_{uid} wants to decrypt the ciphertext CT. If S_{uid} satisfies access structure (M, ρ), the DC selects $\{c_i \in \mathbb{Z}_q^*\}_{i\in\eta}$, such that $\sum_{i\in\eta} c_i M_i = (1, 0, 0, \cdots, 0)$, where $\eta = \{i : \rho(i) \in S_{uid}\}$, and computes

$$m = C_0 - \sum_{aid_k \in I_A} C_{3,i} \cdot K_{uid,aid_k} - c_i n_A \sum_{aid_k \in I_A} \sum_{i\in\eta} (C_{1,i} - C_{2,i} \cdot D_{x_{aid_k},uid})$$

Correctness

$$\begin{aligned} & C_0 - \sum_{aid_k \in I_A} C_{3,i} \cdot K_{uid,aid_k} \\ & + c_i n_A \sum_{aid_k \in I_A} \sum_{i\in\eta} (C_{1,i} - C_{2,i} \cdot D_{x_{aid_k},uid}) \\ & = C_0 - \sum_{aid_k \in I_A} (s\beta_{aid_k} P \cdot (\alpha_{aid_k}/\beta_{aid_k} + 1/\beta_{aid_k})) \\ & + c_i n_A \sum_{aid_k \in I_A} \sum_{i\in\eta} (\lambda_i P + \omega_i PK_{\rho(i)} \\ & - \omega_i \beta_{aid_k} P \cdot (\upsilon_{x_{aid_k}} H_1(x_{aid_k}) + H_1(uid)/\beta_{aid_k})) \\ & = C_0 - \sum_{aid_k \in I_A} (s\alpha_{aid_k} P + sP) \\ & + c_i n_A \sum_{aid_k \in I_A} \sum_{i\in\eta} (\lambda_i P - \omega_i H_1(uid) P) \\ & = C_0 - s \sum_{aid_k \in I_A} \alpha_{aid_k} P - sn_A P \\ & + c_i n_A \sum_{aid_k \in I_A} \sum_{i\in\eta} (\lambda_i P - \omega_i H_1(uid) P) \\ & = C_0 - s \sum_{aid_k \in I_A} \alpha_{aid_k} P - sn_A P + sn_A P \\ & = m \end{aligned}$$

It is worth noting that $\vec{\lambda} \cdot (1, 0, \cdots, 0) = s$, $\vec{\omega} \cdot (1, 0, \cdots, 0) = 0$.

E. ATTRIBUTE REVOKED

If an attribute $\tilde{x}_{aid'}$ is revoked from DC with identity uid, the phase of attribute revoked is as follows:

- *UKeyGen*: The corresponding $AA_{aid'}$ containing the revoked attribute $\tilde{x}_{aid'}$ randomly selects a new version key $VK'_{\tilde{x}_{aid'}} = \upsilon'_{\tilde{x}_{aid'}} (\upsilon'_{\tilde{x}_{aid'}} \neq \upsilon_{\tilde{x}_{aid'}}) \in \mathbb{Z}_q^*$, computes the new public key $\widetilde{PK}_{\tilde{x}_{aid'}} = \beta_{aid'} \upsilon'_{\tilde{x}_{aid'}} H_1(\tilde{x}_{aid'}) P$ and the update key $UK_{\tilde{x}_{aid'}} = \beta_{aid'} (\upsilon'_{\tilde{x}_{aid'}} - \upsilon_{\tilde{x}_{aid'}}) H_1(\tilde{x}_{aid'}) P$. The $AA_{aid'}$ sends the update key $UK_{\tilde{x}_{aid'}}$ to the non-revoked DCs and CSP via the secure channel.
- *SKUpdate*: After receiving the update key $UK_{\tilde{x}_{aid'}}$, the DC computes new secret key

$$\begin{aligned} & \widetilde{SK}_{uid,aid'} \\ & = \{\tilde{K}_{uid,aid'} = K_{uid,aid'}, \\ & \tilde{K}_{\tilde{x}_{aid'},uid} = K_{\tilde{x}_{aid'},uid} + UK_{\tilde{x}_{aid'}}, \\ & , \tilde{K}_{x_{aid'},uid} = K_{x_{aid'},uid}, x_{aid'} \in S_{uid,aid'} \backslash \{\tilde{x}_{aid'}\}\} \end{aligned}$$

- *CTUpdate*: After receiving the update key $UK_{\tilde{x}_{aid'}}$, the CSP computes new ciphertext

$$\begin{aligned} \widetilde{CT} = \{ & \tilde{C}_0 = C_0, \tilde{C}_{2,i} = C_{2,i}, \tilde{C}_{3,i} = C_{3,i}, \\ & \tilde{C}_{1,i} = C_{1,i} + C_{2,i} \cdot UK_{\rho(i)}, \rho(i) = \tilde{x}_{aid'}, \\ & \tilde{C}_{1,i} = C_{1,i}, \rho(i) \neq \tilde{x}_{aid'}\}, i=[1, l] \end{aligned}$$

In the proposed RMA-ABE scheme, only the ciphertext related to revoked attributes is required to update, the other ciphertext will not change.

V. SECURITY ANALYSIS

A. SECURITY PROOF

Theorem 1: The proposed RMA-ABE scheme is IDN-CPA in the standard model under the DDH assumption.

Proof: Even if a polynomial-time adversary $\mathcal{A}$ can break the RMA-ABE scheme with a non-negligible advantage ε, an algorithm $\mathcal{B}$ could be served to settle DDH problem with advantage ε'. Given an instance of DDH problem (P, aP, bP, Z), the target of $\mathcal{B}$ is to distinguish $Z = abP$ or $Z = R$. $\mathcal{B}$ and $\mathcal{A}$ play the following interactive game.

Initialization: $\mathcal{A}$ defines a set of corrupted authorities $U'_A \subseteq U_A$ and a challenge access structure (M^*, ρ^*), then transmits them to $\mathcal{B}$.

Setup: $\mathcal{B}$ selects a hash function $H_1 : \{0, 1\}^* \to \mathbb{Z}_q^*$, returns the system parameters $PP = \{q, P, \mathbb{G}, H_1\}$ to $\mathcal{A}$. In addition, for each uncorrupted authorities in $U_A - U'_A$, $\mathcal{B}$ randomly chooses $\alpha_{aid}, \beta_{aid} \in \mathbb{Z}_q^*$ as the secret keys of AA_{aid}, and computes the public keys $\alpha_{aid}P, \beta_{aid}P$. For each attribute $x_{aid} \in S_{aid}$, $\mathcal{B}$ chooses a version key $VK_{x_{aid}} = \upsilon_{x_{aid}} \in \mathbb{Z}_q^*$, computes public key $PK_{x_{aid}} = \beta_{aid} \upsilon_{x_{aid}} H_1(x_{aid}) aP$. For uncorrupted authorities in $U_A - U'_A$, $\mathcal{B}$ only returns $\{PK_{aid_k}\}_{aid_k \in U_A - U'_A}, \{PK_{x_{aid_k}}\}_{aid_k \in U_A - U'_A}$ to $\mathcal{A}$.

Phase 1: The following queries are made by the adversary $\mathcal{A}$:

- *SKey query*: $\mathcal{A}$ makes secret keys queries on $(uid, S_{uid,aid_k})_{aid_k \in U_A - U'_A}$, $\mathcal{B}$ computes
  $SK_{uid,aid} = \{K_{uid,aid} = \alpha_{aid}/\beta_{aid} + 1/\beta_{aid}$,
  $K_{x_{aid,uid}} = \upsilon_{x_{aid}} a H_1(x_{aid}) + H_1(uid)/\beta_{aid}$,
  $x_{aid} \in S_{uid,aid}\}$
  $\mathcal{B}$ returns $\{SK_{uid,aid_k}\}_{aid_k \in U_A - U'_A}$ to $\mathcal{A}$.
- *UKey query*: $\mathcal{A}$ makes update key queries on S_R. If the attribute $\tilde{x}_{aid} \in S_R$, $\mathcal{B}$ selects a new version key $VK'_{\tilde{x}_{aid}} = \upsilon'_{\tilde{x}_{aid}} (\upsilon'_{\tilde{x}_{aid}} \neq \upsilon_{\tilde{x}_{aid}}) \in \mathbb{Z}_q^*$, computes update key $UK_{\tilde{x}_{aid}} = \beta_{aid} (\upsilon'_{\tilde{x}_{aid}} - \upsilon_{\tilde{x}_{aid}}) H_1(\tilde{x}_{aid}) aP$. $\mathcal{B}$ returns $UK_{\tilde{x}_{aid}}$ to $\mathcal{A}$.

Challenge: $\mathcal{A}$ submits two challenge plaintexts m_0^* and m_1^* to $\mathcal{B}$. $\mathcal{B}$ chooses $s \in \mathbb{Z}_q^*$ and two random vectors $\vec{\lambda} = (s, y_2, y_3, \cdots, y_l), \vec{\omega} = (0, z_2, z_3, \cdots, z_l)$, and calculates $\lambda_i = M_i^* \cdot \vec{\lambda}, \omega_i = M_i^* \cdot \vec{\omega}$, where $i = 1, 2, \cdots, l$. $\mathcal{B}$ flips a bit β and computes

$$C_0^* = m_\beta^* + s \sum_{aid_k \in I_A} \alpha_{aid_k} P,$$

$$C_{1,i}^* = \lambda_i P + \omega_i \beta_{aid_k} \upsilon_{x_{aid_k}} H_1(\rho^*(i)) Z,$$

$$C^*_{2,i} = \omega_i \beta_{aid_k} bP,$$
$$C^*_{3,i} = s\beta_{aid_k} P.$$

$\mathcal{B}$ returns the ciphertext $CT^* = \{C^*_0, (C^*_{1,i}, C^*_{2,i}, C^*_{3,i})_{i=[1,l]}\}$ to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ can issue a number of secret keys or update keys queries as in Phase 1, but the secret keys or update keys relevant to an attributes set satisfying the access structure (M^*, ρ^*) can't be issued by $\mathcal{A}$.

Guess: When $\mathcal{A}$ makes a guess about b, guesses b' is acquired. If $b' = b$, $\mathcal{B}$ outputs 1 guessing $Z = abP$; otherwise, $\mathcal{B}$ outputs 0 indicating $Z = R$.

- When $Z = abP$, it means that it is a valid ciphertext, and the advantage of $\mathcal{A}$ is ε. Thus

$$\Pr[\mathcal{B}(P, aP, bP, Z = abP] = \frac{1}{2} + \varepsilon.$$

- When $Z = R$, it implies that it is absolutely random from $\mathcal{A}$'s perspective. Therefore,

$$\Pr[\mathcal{B}(P, aP, bP, Z = R)] = \frac{1}{2}.$$

Hence, the advantage of $\mathcal{B}$ capable of tackling the DDH problem is at least

$$\begin{aligned}
&\frac{1}{2}(\Pr[\mathcal{B}(P, aP, bP, Z = abP)] \\
&\quad + \Pr[\mathcal{B}(P, aP, bP, Z = R)]) - \frac{1}{2} \\
&= \frac{1}{2}(\frac{1}{2} + \varepsilon + \frac{1}{2}) - \frac{1}{2} \\
&= \frac{\varepsilon}{2}
\end{aligned}$$

B. COLLISION RESISTANCE

The ciphertext may be decrypted by multiple illegal DCs, because they can share own secret keys with each other, where none of them have the secret keys that can decrypt the ciphertext independently. Therefore, for protecting the security of the system, the collusion resistance must be satisfied. We tie the DC's identity *uid* to the secret keys of attributes when the DC requests secret keys, so that they cannot be successfully combined with other DC's secret keys in decryption. Because Alice and Bob have two different uid, $H_1(uid_{Alice}) \neq H_1(uid_{Bob})$, $\sum_{i\in l} c_i \omega_i H_1(uid)P$ cannot be ignored. So, the proposed MA-ABE scheme achieves collusion resistant.

C. FORWARD SECRECY

If an attribute $\tilde{x}_{aid'}$ can be revoked from a DC, the corresponding $PK_{\tilde{x}_{aid'}}$, $SK_{uid,aid'}$ will be updated and transmitted them to the non-revoked DC. Besides, CSP are going to re-encrypted the relevant ciphertext parts under the new $\widetilde{PK}_{\tilde{x}_{aid'}}$. In this way, the new ciphertext cannot be decrypted by previous $SK_{uid,aid'}$. Thus, the forward secrecy can be guaranteed.

VI. PERFORMANCE EVALUATION

This section concerns a performance of the proposed RMA-ABE scheme, then compares it to the related schemes [11], [13]–[23] in light of the functionality, the computation and storage cost.

For fair comparison, the proposed RMA-ABE scheme is compared with schemes [11], [13]–[23] at the same 80-bits security level. Schemes [11], [13]–[23] adopt the symmetric bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$, in which $\mathbb{G}_1$ represents the multiplicative cyclic group with order q by generator g. $\mathbb{G}_1$ is on the basis of a super singular elliptic curve $E : y^2 = x^3 + x \bmod p$. q, p stand for 160-bits and 512-bits prime numbers respectively, in addition, the equation $q \cdot 12 \cdot r = p+1$ can be satisfied. The proposed RMA-ABE scheme adopts an addition cyclic group $\mathbb{G}$ generated by a point P, and the order is q. $\mathbb{G}$ is on the basis of a non-singular elliptic curve $E : y^2 = x^3 + ax + b \bmod p$, where p, q, b stand for 160-bits prime numbers and $a = -3$.

A. FUNCTIONALITY COMPARISON

We compare the functionality of the proposed RMA-ABE scheme and other schemes [11], [13]–[23]. Table 2 summarizes the result of functionality comparison, where ✓ denotes "satisfy" and ✗ denotes "not satisfy".

As shown in Table 2, these schemes [14], [19], [20], [22], [23] support LSSS access structure, multi-authority and attribute revocation, but them can't realize pairing free. The schemes [13], [17], [18] only support LSSS access structure and multi-authority. The access structure used in these schemes [15], [16] is a threshold structure that only supports simple predicates. The scheme [11] only supports the tree as access policy. The scheme [21] adopt the tree as access policy, and realize the attribute revocation. The scheme [25] supports LSSS access structure, pairing-free and attribute revocation, but it can't realize multi-authority. The scheme [26] supports Tree access structure, pairing-free and attribute revocation, but it can't realize multi-authority. In the above schemes, only the proposed scheme uses LSSS access structure to realize MA-ABE without pairing operation, and supports attribute revocation.

B. COMPUTATION COSTS

The performance evaluation contains computation cost comparison between the proposed RMA-ABE scheme and the related schemes [11], [13]–[23]. Let T_p denote the time of a bilinear pairing operation, T_e, T_m, T_{me} represent the time of a scalar multiplication operation in $\mathbb{G}_1$, a scalar multiplication in bilinear pairing and a scalar multiplication in $\mathbb{G}$, respectively. In the comparison, beyond consideration are some lightweight operations, for example, point addition and one-way hash function. The simulation was conducted on desktop computer (i5-M60, 2.53GHz, i5 CPU, 4 GB memory and 64-bit windows10 operating system). Table 3 lists the average runtime of the operations.

Suppose there are k attribute authorities, and the attributes number for the DC obtaining from each attribute authority

TABLE 2. Comparison of functionality.

Schemes	Access structure	Multi-authority	Pairing free	Attribute revoked	Decentralized
Chase *et al.*'s scheme [11]	Tree	✓	✗	✗	✓
Zhang *et al.*'s scheme [13]	LSSS	✓	✗	✗	✓
Yang *et al.*'s scheme [14]	LSSS	✓	✗	✓	✓
Li *et al.*'s scheme [15]	Threshold policy	✓	✗	✗	✓
Xu *et al.*'s scheme [16]	Threshold policy	✓	✗	✗	✓
Belguith *et al.*'s scheme [17]	LSSS	✓	✗	✗	✓
Sethi *et al.*'s scheme [18]	LSSS	✓	✗	✗	✓
Ruj *et al.*'s scheme [19]	LSSS	✓	✗	✓	✓
Yang *et al.*'s scheme [20]	LSSS	✓	✗	✓	✓
Li *et al.*'s scheme [21]	Tree	✓	✗	✓	✓
Liu *et al.*'s scheme [22]	LSSS	✓	✗	✓	✓
Fan *et al.*'s scheme [23]	LSSS	✓	✗	✓	✓
Ding *et al.*'s scheme [25]	LSSS	✗	✓	✓	✓
Wang *et al.*'s scheme [26]	Tree	✗	✓	✓	✓
The proposed scheme	LSSS	✓	✓	✓	✓

TABLE 3. Execution time of operation (millisecond).

Notations	Descriptions	Execution time
T_p	Bilinear pairing operation	10.3092
T_e	Scalar multiplication operation in $\mathbb{G}_1$	0.5200
T_m	Scalar multiplication operation in $\mathbb{G}_T$	1.4202
T_{me}	Scalar multiplication operation in $\mathbb{G}$	0.3851

are same, the number is set as j. Since the schemes [25], [26] are single-authority, the simulation result of computation costs and storage overhead are one-order equation, the schemes [11], [13]–[23] and the proposed scheme are multi-authority, the simulation result of computation costs and storage overhead are binary one-order equation, the performance analysis should be discussed separately.

Based on the experiment result, the computation overhead of the proposed scheme and [11], [13]–[23] are summarized in Table 4, where "/" denotes that the scheme does not involve this feature.

The computation time of encryption, decryption and ciphertext update phase is shown in Figure 2. Figure 2(a) and 2(b) list the computation time on encryption and decryption corresponding to the number of AAs respectively, with the attribute number j involves from each AA set as 10. From Figure 2(a) and 2(b), it can be seen that the computation cost on the encryption and decryption phases rise with the number of AAs linearly. As far as Figure 2(a), for $k = 10$, the encryption computation cost of the proposed scheme as well as the schemes [13]–[23] is equal to 118, 261, 402, 158, 3374, 272, 1250, 2174, 273, 171, 2330 and 1310 ms. Compared with the schemes [13]–[23], the proposed scheme saves 55%, 71%, 25%, 96%, 57%, 90%, 94%, 57%, 31%, 95%, 91% of the computing cost respectively. In Figure 2(b), for $k = 10$, the decryption computation overhead of the proposed scheme and the schemes [11], [13]–[23] respectively is equal to 156, 3091, 4121, 5150, 2184, 5163, 2068, 5151, 2061, 3436, 2164, 3150 and 6191 ms. Compared with the schemes [11], [13]–[23], the proposed scheme saves 95%, 96%, 97%, 93%, 97%, 92%, 97%, 92%, 95%, 93%, 95%, 97% of the computing cost respectively.

Figure 2(c) and 2(d) depict the comparison of encryption and decryption time versus the number of attributes in per AA, in which the number of AAs is fixed at 10. From Figure 2(c) and 2(d), the computation cost rise with the number of attributes linearly in per AA in the encryption and decryption phases of all the schemes. In Figure 2(c), for $j = 10$, the encryption computation cost of the proposed scheme as well as the schemes [13]–[23] respectively is equal to 118, 261, 402, 158, 3374, 272, 1250, 2174, 273, 171, 2330 and 1310 ms. Compared with the schemes [13]–[23], the proposed scheme respectively saves 55%, 71%, 25%, 96%, 57%, 90%, 94%, 57%, 31%, 95%, 91% of the computing cost. In Figure 2(d), for $j = 10$, the decryption computation overhead of the proposed scheme and the schemes [11], [13]–[23] respectively is equal to 156, 3091, 4121, 5150, 2184, 5163, 2068, 5151, 2061, 3436, 2164, 3150 and 6191 ms. Compared with the schemes [11], [13]–[23], the proposed scheme saves 95%, 96%, 97%, 93%, 97%, 92%, 97%, 92%, 95%, 93%, 95%, 97% of the computing cost respectively. Because the schemes [25], [26] are single authority, for $k = 10$, the encryption computation cost of the schemes [25], [26] respectively is equal to 11.9381 ms and 12.3232 ms. The decryption computation cost of the schemes [25], [26] respectively is equal to 23.106 ms and 19.6401 ms. Suppose there exist 10 attribute authorities, the encryption computation cost of the schemes [25], [26] respectively is equal to 119.381 ms and 123.232 ms, the decryption computation cost of the schemes [25], [26] respectively is equal to 231.06 ms and 196.401 ms. The encryption and decryption time of the proposed scheme is respectively equal to 117.8 ms and 155.8 ms, it is easy to recognize that the computation overhead of the proposed scheme is the lowest.

Figure 2(e) shows the comparison of ciphertext update time corresponding to the number of the revoked attributes. If $n_R = 10$, the computation cost of the proposed scheme and the schemes [14], [19]–[23] in the ciphertext update phase respectively is equal to 3.8 5.2, 103.1, 10.4, 10.4, 14.2, and 5.2 ms. Compared with the schemes [14], [19]–[23], the proposed scheme saves 27%, 96%, 63%, 63%, 73%, 27% of the computing cost respectively.

TABLE 4. Computation overhead.

Schemes	Encryption	Decryption	Ciphertext update
Chase *et al.*'s scheme [11]	$T_p + (kj+1)T_e \approx 0.52kj + 10.8$	$3kjT_p + T_m \approx 30.9kj + 1.42$	/
Zhang *et al.*'s scheme [13]	$(5kj+2)T_e \approx 2.6kj + 1.04$	$4kjT_p + T_m \approx 41.2kj + 1.42$	/
Yang *et al.*'s scheme [14]	$kT_m + (4kj+k+1)T_e \approx 2.08kj + 1.94k + 0.52$	$5kjT_p + T_e \approx 51.5kj + 0.52$	$n_R T_e \approx 0.52 n_R$
Li *et al.*'s scheme [15]	$T_m + (3kj+1)T_e \approx 1.56kj + 1.96$	$2kjT_p + 2kT_m + 2kjT_e \approx 21.66kj + 2.84k$	/
Xu *et al.*'s scheme [16]	$(3kj+2)T_p + 5kjT_e \approx 33.53kj + 20.62$	$(5kj+1)T_p + 2T_m \approx 51.5kj + 13.2$	/
Belguith *et al.*'s scheme [17]	$T_p + T_m + (5kj+1)T_e \approx 2.6kj + 12.3$	$2kjT_p + 3T_m + 3T_e \approx 20.62kj + 6$	/
Sethi *et al.*'s scheme [18]	$(kj+1)T_p + 4kjT_e \approx 12.4kj + 10.31$	$5kjT_p + T_m \approx 51.5kj + 1.42$	/
Ruj *et al.*'s scheme [19]	$(2kj+1)T_p + 2kjT_e \approx 21.64kj + 10.31$	$2kjT_p + T_m \approx 20.6kj + 1.42$	$n_R T_p \approx 10.31 n_R$
Yang *et al.*'s scheme [20]	$kT_m + (5kj+2)T_e \approx 2.6kj + 1.24k + 1.04$	$(3kj+2k)T_p + kjT_m \approx 32.3kj + 20.6k$	$2n_R T_e \approx 1.04 n_R$
Li *et al.*'s scheme [21]	$kT_m + (3kj+1)T_e \approx 1.56kj + 1.42k + 0.52$	$(2kj+k)T_p + T_m \approx 20.6kj + 10.3k + 1.42$	$2n_R T_e \approx 1.04 n_R$
Liu *et al.*'s scheme [22]	$(2kj+1)T_p + 5kjT_e \approx 23.2kj + 10.31$	$3kjT_p + kjT_e \approx 31.5kj$	$n_R T_m \approx 1.42 n_R$
Fan *et al.*'s scheme [23]	$(kj+1)T_p + 5kjT_e \approx 13kj + 10.31$	$6kjT_p + T_m \approx 61.9kj + 1.42$	$n_R T_e \approx 0.52 n_R$
Ding *et al.*'s scheme [25]	$(3k+1)T_{me} \approx 1.1553k + 0.3851$	$6kT_{me} \approx 2.3106k$	/
Wang *et al.*'s scheme [26]	$(3k+2)T_{me} \approx 1.1553k + 0.7702$	$(5k+1)T_{me} \approx 1.9255k + 0.3851$	$2n_R T_{me} \approx 0.7702 n_R$
The proposed scheme	$(3kj+k)T_{me} \approx 1.14kj + 0.38k$	$(4kj+k)T_{me} \approx 1.52kj + 0.38k$	$n_R T_{me} \approx 0.38 n_R$

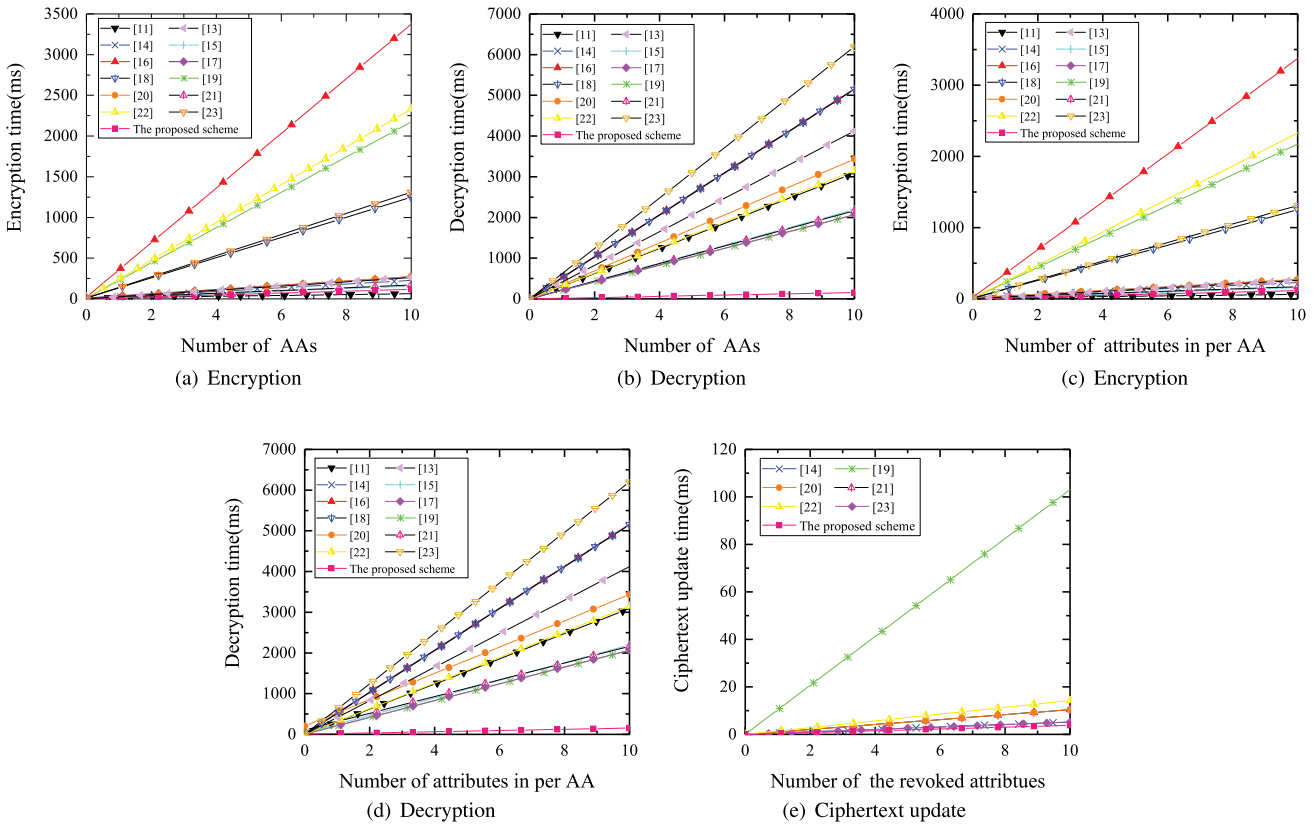

FIGURE 2. Comparison of encryption, decryption and ciphertext update time.

When an attribute is revoked, although the proposed scheme is supposed to send update key to each user, the computation cost of the proposed scheme is lower compared with those of related schemes [20]–[23], [25], [26].

According to Figure 2(a)-(e), we can conclude that the computation cost of the proposed scheme is the lowest.

C. STORAGE OVERHEAD

This subsection gives the comparison in terms of the storage overhead between the proposed scheme and the related schemes [11], [13]–[23]. The length of elements in $|\mathbb{G}_1|$, $|\mathbb{G}_T|$, $|\mathbb{G}|$, $|\mathbb{Z}_q|$ respectively are 512 bits, 1024 bits, 160 bits and 160 bits. The comparison of storage overhead is shown in Table 5.

Figure 3 describes the comparison of storage cost of secret key, ciphertext and update key respectively, where the involved attribute number j from each AA is set as 5. In Figure 3(a), for $k = 5$, the secret key size of the proposed scheme and the schemes [11], [13]–[23] respectively is equal to 4800, 15360, 26112, 20480, 17920, 12800, 25600, 26112, 12800, 17920, 17920, 25600, and 13312 bits. Compared with the schemes [11], [13]–[23], the proposed scheme saves 67%, 81%, 75%, 72%, 61%, 81%, 81%, 61%, 72%, 72%, 80%, 62% respectively. Likewise, in Figure 3(b), for

TABLE 5. Storage overhead.

Schemes	Secret key size	Ciphertext size	Update key size
Chase *et al.*'s scheme [11]	$(kj+k)\|\mathbb{G}_1\| \approx 512kj+512k$	$(kj+1)\|\mathbb{G}_1\|+\|\mathbb{G}_T\| \approx 512kj+1536$	/
Zhang *et al.*'s scheme [13]	$(2kj+2)\|\mathbb{G}_1\| \approx 1024kj+512$	$(3kj+1)\|\mathbb{G}_1\|+\|\mathbb{G}_T\| \approx 1536kj+1536$	/
Yang *et al.*'s scheme [14]	$(kj+3k)\|\mathbb{G}_1\| \approx 512kj+1536k$	$(3kj+k+1)\|\mathbb{G}_1\|+\|\mathbb{G}_T\| \approx 1536kj+512k+1536$	$n_R\|\mathbb{G}_1\| \approx 512n_R$
Li *et al.*'s scheme [15]	$(kj+2k)\|\mathbb{G}_1\| \approx 512kj+1024k$	$(2kj+1)\|\mathbb{G}_1\|+\|\mathbb{G}_T\| \approx 1024kj+1536$	/
Xu *et al.*'s scheme [16]	$kj\|\mathbb{G}_1\| \approx 512kj$	$5kj\|\mathbb{G}_1\|+(kj+1)\|\mathbb{G}_T\| \approx 3584kj+1024$	/
Belguith *et al.*'s scheme [17]	$2kj\|\mathbb{G}_1\| \approx 1024kj$	$(3kj+1)\|\mathbb{G}_1\|+\|\mathbb{G}_T\| \approx 2560kj+1536$	/
Sethi *et al.*'s scheme [18]	$(2kj+1)\|\mathbb{G}_1\| \approx 1024kj+512$	$4kj\|\mathbb{G}_1\|+(kj+1)\|\mathbb{G}_T\| \approx 3072kj+1024$	/
Ruj *et al.*'s scheme [19]	$kj\|\mathbb{G}_1\| \approx 512kj$	$2kj\|\mathbb{G}_1\|+(kj+1)\|\mathbb{G}_T\| \approx 2048kj+1024$	$n_R\|\mathbb{G}_1\|+\|\mathbb{G}_T\| \approx 512n_R+1024$
Yang *et al.*'s scheme [20]	$(kj+2k)\|\mathbb{G}_1\| \approx 512kj+1024k$	$(4kj+2)\|\mathbb{G}_1\|+\|\mathbb{G}_T\| \approx 2048kj+2048$	$2n_R\|\mathbb{G}_1\| \approx 1024n_R$
Li *et al.*'s scheme [21]	$(kj+2k)\|\mathbb{G}_1\| \approx 512kj+1024k$	$(3kj+1)\|\mathbb{G}_1\|+\|\mathbb{G}_T\| \approx 1536kj+2048$	$2n_R\|\mathbb{G}_1\| \approx 1024n_R$
Liu *et al.*'s scheme [22]	$2kj\|\mathbb{G}_1\| \approx 1024kj$	$3kj\|\mathbb{G}_1\|+(kj+1)\|\mathbb{G}_T\| \approx 2560kj+1024$	$n_R\|\mathbb{G}_1\| \approx 512n_R$
Fan *et al.*'s scheme [23]	$(kj+1)\|\mathbb{G}_1\| \approx 512kj+512$	$6kj\|\mathbb{G}_1\|+\|\mathbb{G}_T\| \approx 3072kj+1024$	$n_R\|\mathbb{G}_1\| \approx 512n_R$
Ding *et al.*'s scheme [25]	$k\|\mathbb{Z}_q\| \approx 160k$	$(2k+1)\|\mathbb{G}\| \approx 320k+160$	/
Wang *et al.*'s scheme [26]	$3k\|\mathbb{Z}_q\| \approx 480k$	$(2+3k)\|\mathbb{G}\| \approx 480k+320$	$2n_R\|\mathbb{G}\| \approx 320n_R$
The proposed scheme	$(kj+k)\|\mathbb{Z}_q\| \approx 160kj+160k$	$(2kj+2)\|\mathbb{G}\| \approx 320kj+320$	$n_R\|\mathbb{G}\| \approx 160n_R$

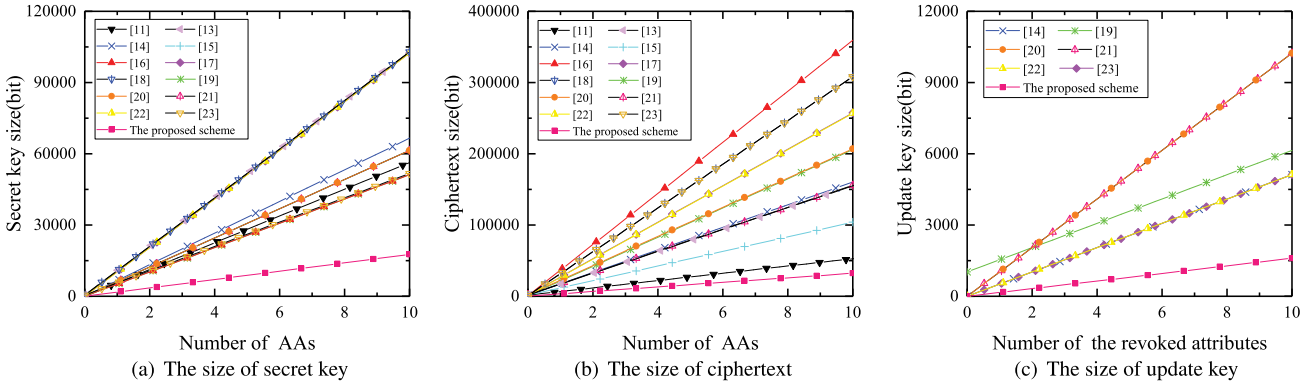

FIGURE 3. Comparison of secret key, ciphertext and update key size.

$k = 5$, the ciphertext size of the proposed scheme and the schemes [11], [13]–[23] respectively is equal to 8320, 14336, 39936, 42496, 27136, 90624, 65536, 77824, 52224, 53248, 40448, 64102, and 77824 bits. Compared with the schemes [11], [13]–[23], the proposed scheme saves 42%, 79%, 80%, 69%, 91%, 87%, 90%, 84%, 84%, 80%, 87%, 89% respectively. In Figure 3(c), for $n_R = 5$, the update key size of the proposed scheme and the schemes [14], [19]–[23] respectively is equal to 800, 2560, 3584, 5120, 5120, 2560, and 2560 bits. Compared with the schemes [14], [19]–[23], the proposed scheme saves 68%, 77%, 84%, 84%, 69%, 69% respectively. Because the schemes [25], [26] are single authority, for $k = 5$, the secret key size of the schemes [25], [26] respectively is equal to 800 bits and 2400 bits, the ciphertext size of the schemes [25], [26] respectively is equal to 1760 bits and 2720 bits, but they are single authority.

So compared with the schemes [11], [13]–[23], the storage overhead of the proposed scheme is the lowest.

VII. CONCLUSION

This paper proposes an efficient RMA-ABE system for cloud storage, which is on the basis of the elliptic curve cryptography. The proposed scheme will not need any bilinear pairing operations any more. The version key is introduced into the attribute to achieve the attribute revocation. Security proof demonstrates that the proposed scheme enjoys the confidentiality. In addition, by using the unique identity *uid* tied to the secret keys of attributes, collusion resistant is realized. It is showed in the performance analysis that the proposed scheme is high-efficiency in storage as well as computation cost.

REFERENCES

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep., Sep. 2009, no. 53, pp. 267–269.

[2] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology–(EUROCRYPT)*. Berlin, Germany: Springer, Jan. 2005, pp. 457–473.

[3] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur. (CCS)*, 2006, pp. 89–98.

[4] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy (SP)*, Berkeley, CA, USA, May 2007, pp. 321–334.

[5] S. Yu, K. Ren, and W. Lou, "FDAC: Toward fine-grained distributed data access control in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 4, pp. 673–686, Apr. 2011.

[6] Z. Wan, J. Liu, and R. H. Deng, "HASBE: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 743–754, Apr. 2012.

[7] K. Yang, X. Jia, and K. Ren, "Secure and verifiable policy update outsourcing for big data access control in the cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 12, pp. 3461–3470, Dec. 2015.

[8] J. Li, X. Lin, Y. Zhang, and J. Han, "KSF-OABE: Outsourced attribute-based encryption with keyword search function for cloud storage," *IEEE Trans. Services Comput.*, vol. 10, no. 5, pp. 715–725, Sep. 2017.

[9] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, "User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1767–1777, Jun. 2018.

[10] M. Chase, "Multi-authority attribute based encryption," in *Proc. Theory Cryptogr. Conf.* Berlin, Germany: Springer, Feb. 2007, pp. 515–534.
[11] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proc. 16th ACM Conf. Comput. Commun. Secur. (CCS)*, New York, NY, USA, 2009, pp. 121–130.
[12] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Advances in Cryptology–(EUROCRYPT)*. Berlin, Germany: Springer, May 2011, pp. 568–588.
[13] Y. Zhang, D. Zheng, Q. Li, J. Li, and H. Li, "Online/offline unbounded multi-authority attribute-based encryption for data sharing in mobile cloud computing," *Secur. Commun. Netw.*, vol. 9, no. 16, pp. 3688–3702, Aug. 2016.
[14] K. Yang, X. Jia, K. Ren, B. Zhang, and R. Xie, "DAC-MACS: Effective data access control for multiauthority cloud storage systems," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 11, pp. 1790–1801, Nov. 2013.
[15] J. Li, X. Huang, X. Chen, and Y. Xiang, "Securely outsourcing attribute-based encryption with checkability," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 8, pp. 1343–1354, Aug. 2014.
[16] Q. Xu, C. Tan, W. Zhu, Y. Xiao, Z. Fan, and F. Cheng, "Decentralized attribute-based conjunctive keyword search scheme with online/offline encryption and outsource decryption for cloud computing," *Future Gener. Comput. Syst.*, vol. 97, pp. 306–326, Aug. 2019.
[17] S. Belguith, N. Kaaniche, M. Laurent, A. Jemai, and R. Attia, "PHOABE: Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted IoT," *Comput. Netw.*, vol. 133, pp. 141–156, Mar. 2018.
[18] K. Sethi, A. Pradhan, and P. Bera, "Practical traceable multi-authority CP-ABE with outsourcing decryption and access policy updation," *J. Inf. Secur. Appl.*, vol. 51, Apr. 2020, Art. no. 102435.
[19] S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: Distributed access control in clouds," in *Proc. IEEE 10th Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Nov. 2011, pp. 91–98.
[20] K. Yang and X. Jia, "Expressive, efficient, and revocable data access control for multi-authority cloud storage," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 8, pp. 2201–2210, Jul. 2014.
[21] Q. Li, J. Ma, R. Li, X. Liu, J. Xiong, and D. Chen, "Secure, efficient and revocable multi-authority access control system in cloud storage," *Comput. Secur.*, vol. 59, pp. 45–59, Jun. 2016.
[22] Z. Liu, Z. L. Jiang, X. Wang, and S. M. Yiu, "Practical attribute-based encryption: Outsourcing decryption, attribute revocation and policy updating," *J. Netw. Comput. Appl.*, vol. 108, pp. 112–123, Apr. 2018.
[23] K. Fan, T. Liu, K. Zhang, H. Li, and Y. Yang, "A secure and efficient outsourced computation on data sharing scheme for privacy computing," *J. Parallel Distrib. Comput.*, vol. 135, pp. 169–176, Jan. 2020.
[24] M. Pirretti, P. Traynor, P. Mcdaniel, and B. Waters, "Secure attribute-based systems," *J. Comput. Secur.*, vol. 18, no. 5, pp. 99–112, Oct. 2010.
[25] S. Ding, C. Li, and H. Li, "A novel efficient pairing-free CP-ABE based on elliptic curve cryptography for IoT," *IEEE Access*, vol. 6, pp. 27336–27345, May 2018.
[26] Y. Wang, B. Chen, L. Li, Q. Ma, H. Li, and D. He, "Efficient and secure ciphertext-policy attribute-based encryption without pairing for cloud-assisted smart grid," *IEEE Access*, vol. 8, pp. 40704–40713, Feb. 2020.
[27] A. Beimel, "Secure schemes for secret sharing and key distribution," Ph.D. dissertation, Israel Inst. Technol., Technion, Haifa, Israel, Tech. Rep., 1996.
[28] V. S. Miller, "Use of elliptic curves in cryptography," in *Advances in Cryptology–(CRYPTO)*. Berlin, Germany: Springer, Aug. 1985, pp. 416–426.

YANG MING (Member, IEEE) received the B.S. and M.S. degrees in mathematics from the Xi'an University of Technology, Xi'an, China, in 2002 and 2005, respectively, and the Ph.D. degree in cryptography from Xidian University, Xi'an, in 2008. He is currently a Professor with Chang'an University. His main research interests include cryptography and wireless network security.

BAOKANG HE received the B.S. degree from the Henan University of Technology, in 2017, and the M.S. degree from Chang'an University, Xi'an, China, in 2020. His main research interests include cryptography and attribute-based encryption.

CHENHAO WANG received the B.S. degree from Dalian University, in 2020. He is currently pursuing the master's degree with Chang'an University, Xi'an, China. His main research interests include cryptography and attribute-based encryption.