IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON MOBILE MULTIMEDIA: METHODOLOGY AND APPLICATIONS

Received August 16, 2019, accepted August 28, 2019, date of publication September 3, 2019,
date of current version September 25, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2939299

Efficient Privacy-Preserving Access Control of Mobile Multimedia Data in Cloud Computing

QI LI[1,2], YOULIANG TIAN[1], YINGHUI ZHANG[3], LIMIN SHEN[4], AND JINGJING GUO[5]

[1]State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang 550025, China
[2]Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China
[3]School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China
[4]School of Computer Science and Technology, Nanjing Normal University, Nanjing 210023, China
[5]School of Cyber Engineering, Xidian University, Xi'an 710071, China

Corresponding author: Youliang Tian (yltian@gzu.edu.cn)

This work was supported in part by the Foundation of Guizhou Provincial Key Laboratory of Public Big Data under Grant 2018BDKFJJ015, in part by the National Key Research and Development Program of China under Grant 2018YFC1314903, in part by the National Nature Science Foundation of China under Grant 61802195 and Grant 61602360, in part by the Key Research and Development Program of Shaanxi under Grant 2019KW-053, and in part by the New Star Team of Xi'an University of Posts and Telecommunications under Grant 2016-02.

ABSTRACT With the nature of allowing the encryptor to define the access policy before encrypting a message, Ciphertext-policy attribute-based encryption (CP-ABE) has been widely adopted as a primitive to design cloud-assisted mobile multimedia data sharing system. However, in most previous works, the access policy sent along with ciphertext remains in the palintext form, which would expose the user's privacy to anyone who can get the ciphertext even if he is not authorized to decrypt. In addition, the resource-constrained mobile devices can not carry out frequent encryption and decryption task caused by CP-ABE. Such drawbacks may reduce the enthusiasm of consumers to share the multimedia data via their mobile devices. In this paper, we proposed PPCMM, a privacy-preserving cloud-assisted mobile multimedia data sharing scheme, where each attribute is described by an attribute name and an attribute value. The attribute values are embedded in the chphertext and only the attribute names are revealed in the access policy. The encryption is divided to two phase: online and offline. In the offline stage, the data owner can prepare the intermediate ciphertext components. Once receiving the encryption requirement of a specific access policy and the multimedia data, the data owner can quickly form the final legal ciphertext in the online stage. By employing the decryption outsourcing technique, most computation overhead of matching test and decryption is offload to the cloud server. The security proof showed that PPCMM is adaptively secure in the standard model. The performance analysis indicated that PPCMM greatly reduces the computation cost in both online encryption and user decryption.

INDEX TERMS Mobile multimedia data, access control, CP-ABE, partially hidden policy, online/offline encryption, efficient decryption.

I. INTRODUCTION

Currently, the rapidly advanced cloud-assisted mobile multimedia services motivate the owner to share their multimedia data via the cloud platform. However, the data security and privacy concerns would arise while enjoying the rich computation and storage resources for saving the overhead of mobile devices. In particularly, once the multimedia data is outsourced to the cloud, the data owner has to rely on the cloud server to enforce the access control. But the cloud server can not be fully trusted and it would acquire the owner's privacy and recommend advertisement endlessly. A well adoptable approach is to encrypt the multimedia data before offloading it to the cloud. Whereas, the adoption of traditional cryptology or identity-based encryption might lead to complicated key distribution and management.

Attribute-based encryption (ABE) [1] has been seemed as a significant solution for enforcing fine-grained access control over encrypted message. Due to the nature of linking the access policy with the data to be encrypted, Ciphertext-policy

The associate editor coordinating the review of this article and approving it for publication was Dapeng Wu.

ABE (CP-ABE) [2] is more appropriate for the data owner to specify and realize flexible access policy. In traditional CP-ABE schemes [2]–[5], the access policy used to encrypt is sent together with the ciphertext in the clear form. It is readable to anyone who obtains the ciphertext no matter his attributes can match the access policy or not. Such process may expose the owner's privacy. For example, a multimedia data owner encrypts his electrocardiogram under a concrete access policy defined as ('department: cardiac surgery' and 'title: archiater'). Then everyone can deduce that this owner is suffering from a heart disease. Thus, it is necessary to conceal the specific access policy from being peeped by any unauthorized user.

With the rapidly progress of mobile communication technology [6]–[10], Internet of things (IOT) [10]–[14] and cloud computing [15]–[17], an increasing number of people are enabled to upload and access the multimedia data anywhere and anytime via their mobile devices. Unfortunately, such mobile devices are inherently lack of sufficient battery capacity and adequate computation resource, which may prevent them from executing frequently encryption or decryption operations of CP-ABE. More precisely, the encryption cost and decryption overhead are commonly caused by complicated bilinear pairings and modular exponential operations. On one hand, the encryption cost is usually linear with the scale of access policy. It is considerable to split the encryption operation into two phases: offline and online. In the offline phase, the owner prepares some intermediate ciphertext components in the charging mode or on another device. Given an access policy of multimedia data in the online phase, the owner can rapidly construct the final ABE ciphertext and significant reduce the battery consumption. On the other hand, the decryption overhead usually scales with the number of involved attributes as well, which is burdensome for the mobile devices with limited computation resources. On the contrary, the cloud server has nearly infinite computation resources. It is desirable to securely outsource the heavy decryption to the cloud server without leaking any sensitive privacy.

Aiming to solve the above mentioned issues, in this paper, we propose PPCMM, an efficient privacy-preserving cloud-assisted mobile multimedia data access control scheme, which simultaneously achieves three significant properties including partially hidden access policy, online/offline encryption and outsourced decryption. Similar to [18], PPCMM requires the matching test method to check if the user's attributes satisfy the partially hidden access policy before decrypting the ciphertext. The main contributions are as follows:

1. **Partially hidden access policy**. Each attribute in PPCMM is composed of two parts: attribute name and attribute value. In the CP-ABE ciphertext, the concrete attribute values of the specific access policy are embedded and hidden. The access policy sent along with the ciphertext only contains the generic attribute names.

2. **Online/Offline encryption**. Different from most previous works which accomplish the whole encryption task in the online phase, PPCMM allows the mobile device to pre-compute at most $\jmath$ unspecified attribute cipohertext components in the offline phase. Once receiving the specific access policy, the final ciphertext can be quickly formed without invoking any complicated bilinear pairing or modular exponential operation.

3. **Outsourced decryption**. The complicated matching test operation and most decryption overhead are outsourced to the cloud without affecting the data confidentiality and the policy privacy. The decryption cost on the user side requires only one modular exponential operation.

4. **Adaptive security and practicability**. We prove the adaptive security of PPCMM in the standard model. The theoretical analysis and experimental results show that PPCMM is efficient and practicable.

II. RELATED WORKS

The notion of ABE was first formalized in [1]. Shortly afterwards, Goyal *et al*. [19] and Bethencourt *et al*. [2] presented the first key-policy ABE (KP-ABE) scheme and CP-ABE scheme, respectively. In contrast to CP-ABE, the user's key is labeled by an access policy and the ciphertext is created basing on an attribute set in KP-ABE. Different from the previous ABE works, Lewko *et al*. [4] firstly achieved the adaptive security in both KP-ABE and CP-ABE, where the adversary is not forced to state the challenge attribute set or access policy before initializing the system parameters. On prime order groups, Rouselakis and Waters [5] presented two large universe ABE schemes, where the attribute universe is exponentially large and no extra bound is imposed on the size of public parameters. Subsequently, various researches [16], [20]–[23] demonstrated how to deploy ABE in practice to protect the data confidentiality and achieve fine-grained access control. Nevertheless, the issues of attribute privacy leakage, efficient online encryption and decryption outsourcing have not been addressed.

Nishide *et al*. [24] first introduced a CP-ABE scheme with partially hidden policy, which is proved to be selectively secure. Whereafter, Lai *et al*. [25] and Jin *et al*. [26] separately constructed an adaptively secure CP-ABE supporting the similar AND-gate access policy as in [24]. To improve the policy expressiveness, Lai *et al*. [27] gave a partially hidden CP-ABE scheme supporting any monotonic policy which can be expressed by a linear secret sharing scheme (LSSS). Zhang *et al*. [18] proposed a CP-ABE scheme with any partially hidden LSSS policy and large universe. Different from the employed matching test before decryption [18], [27], Zhang *et al*. [28] developed a new verification method to check if the user possesses appropriate attributes that match the partially hidden access policy. However, their work assumes that the match is true before obtaining the final check result, which may cause some undesired computation cost even if it requires only two bilinear pairings.

Hohenberger and Waters [29] first brought the online/offline technique into ABE framework. In their basic CP-ABE scheme, an intermediate ciphertext which consists of at most $\jmath$ attribute ciphertext elements are created in the offline phase. Each element can be regarded as an encryption of an unspecified attribute with a share of a random secret. In the online phase, once given a specific access policy, the encryptor can rapidly form the final legal ABE ciphertext. In [30], Zhang *et al*. proposed an online/offline and large universe multi-authority CP-ABE scheme. These two schemes [29], [30] are both proved in the selective model.

To alleviate the heavy decryption overhead of users, Green *et al*. [31] studied the decryption outsourcing method for ABE to leverage the abundant computation resources of cloud. In [32], Li *et al*. designed an attribute-based access control scheme in cloud storage with multi-authority and oursourced decryption. Ning *et al*. [33] proposed a verifiability method to check if the cloud correctly performs the outsourced decryption. However, the access policy in these CP-ABE schemes [31]–[33] is clear to anyone.

III. SYSTEM MODEL, DEFINITION AND SECURITY MODEL

A. SYSTEM MODEL

Fig. 1 depicts the system model of PPCMM, involving four distinct entities: attribute authority (AA), cloud service provider (CSP), multimedia data owner (MDO) and multimedia data user (MDU).

AA: It is in charge of initializing the system, publishing the public parameters and generating the keys for MDU according to his attribtues.

CSP: It stores the encrypted multimedia data and the corresponding ABE ciphertext. It also provides computation outsourcing service of matching test and partial decryption.

MDO: It owns the multimedia data to be shared via CSP. Before encrypting and uploading the multimedia data to CSP, it has to define an access policy where each attribute is described by an attribute name and the corresponding attribute value. To save the battery power, MDO can prepare some intermediate ciphertext components before executing the online encryption operation on a specific access policy and a concrete multimedia data file.

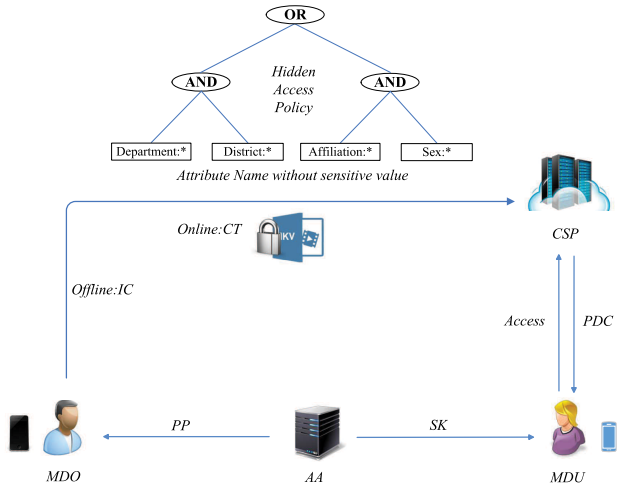


FIGURE 1. System Model of PPCMM.

MDU: It obtains the private keys from AA and wants to access the encrypted multimedia data in CSP. The access succeeds only if both the attribute names and the correspond values match the embedded access policy. MDU can call CSP to perform the matching test and partial decryption operation.

In PPCMM, AA is fully trusted. CSP is honest-but-curious. That is, CSP honestly executes the assigned procedures. But it might try to find out the sensitive information of encrypted multimedia data as much as possible. The malicious MDUs may combine their keys to obtain the access right of the ciphertext that none of them is authorized to decrypt.

B. DEFINITION OF PPCMM

The proposed PPCMM consists of the following 8 algorithms.

- $\mathsf{Setup}(1^\lambda) \rightarrow (\mathsf{SP}, \mathsf{MSK})$: This algorithm takes in a security parameter λ. It then outputs the system public parameters SP and the master secret key MSK.
- $\mathsf{Offline\ Encryption}(\mathsf{SP}) \rightarrow (\mathsf{IC})$: This algorithm takes in SP and outputs a intermediate ciphertext IC.
- $\mathsf{Online\ Encryption}(\mathbb{A}, \mathsf{SP}, \mathsf{IC}) \rightarrow (\mathsf{CT}_{\mathbb{A}})$: This algorithm takes in a specific access structure $\mathbb{A} = (\mathbf{A}, \rho, \mathcal{T})$, SP and IC. It then outputs a session key Key and a ciphertext $\mathsf{CT}_{\mathbb{A}}$.
- $\mathsf{KeyGen}(\mathsf{SP}, \mathsf{MSK}, \mathcal{S}) \rightarrow (\mathsf{UDK}, \mathsf{SK}_{\mathcal{S}})$: This algorithm takes in SP, MSK and an attribute set $\mathcal{S}$. It outputs a user decryption key UDK and the corresponding intermediate key $\mathsf{SK}_{\mathcal{S}}$.
- $\mathsf{Ciphertext\ Transmission}(\mathsf{SP}, \mathsf{UDK}, \mathsf{CT}_{\mathbb{A}}) \rightarrow (\mathsf{CT}'_{\mathbb{A}})$: This algorithm takes in SP, UDK and $\mathsf{CT}_{\mathbb{A}}$. It outputs a transmission ciphertext $\mathsf{CT}'_{\mathbb{A}}$.
- $\mathsf{Matching\ Test}(\mathsf{SP}, \mathsf{SK}_{\mathcal{S}}, \mathsf{CT}'_{\mathbb{A}}) \rightarrow (1 \text{ or } \perp)$: This algorithm takes in SP, $\mathsf{SK}_{\mathcal{S}}$ and $\mathsf{CT}'_{\mathbb{A}}$. If $\mathcal{S}$ matches $\mathbb{A}$, it outputs 1 and calls the $\mathsf{Partial\ Decryption}$ algorithm. Otherwise, it outputs $\perp$.
- $\mathsf{Partial\ Decryption}\ (\mathsf{SP}, \mathsf{SK}_{\mathcal{S}}, \mathsf{CT}'_{\mathbb{A}}) \rightarrow (\mathsf{PDC})$: This algorithm takes in SP, $\mathsf{SK}_{\mathcal{S}}$ and $\mathsf{CT}'_{\mathbb{A}}$. It outputs a partial decryption ciphertext PDC.
- $\mathsf{User\ Decryption}\ (\mathsf{SP}, \mathsf{UDK}, \mathsf{PDC}) \rightarrow (\mathsf{Key})$: This algorithm takes in SP, UDK and PDC. It returns Key.

C. SECURITY MODEL

The security model is described by the following game between an adversary $\mathcal{A}$ and a simulator $\mathcal{B}$.

- **Setup.** $\mathcal{B}$ performs the setup algorithm and sends SP to $\mathcal{A}$.
- **Phase 1.** $\mathcal{B}$ initializes an integer counter $\hbar = 0$, an empty table T and an empty set D. $\mathcal{A}$ could adaptively make the key queries as follows:
  - $\mathsf{Create}(\mathcal{S})$: $\mathcal{B}$ sets $\hbar := \hbar + 1$. It performs KeyGen on the queried attribute set $\mathcal{S}$ to get UDK and $\mathsf{SK}_{\mathcal{S}}$. It then puts $(\hbar, \mathcal{S}, \mathsf{UDK}, \mathsf{SK}_{\mathcal{S}})$ in T.

- **Corrupt**(i): $\mathcal{B}$ checks if the i-th entry $(\hbar, \mathcal{S}, \mathsf{UDK}, \mathsf{SK}_{\mathcal{S}})$ can be found in T. If so, it sets $D := D \bigcup \{\mathcal{S}\}$ and outputs $(\mathsf{UDK}, \mathsf{SK}_{\mathcal{S}})$. Otherwise, it outputs $\perp$.

- **Challenge.** $\mathcal{A}$ submits $\mathbb{A}^*$ as the challenge access structure under the restriction that none of the elements in D can match $\mathbb{A}^*$. $\mathcal{B}$ performs **Offline Encryption** and **Online Encryption** to get $(\mathsf{Key}^*, \mathsf{CT}^*_{\mathbb{A}^*})$. It then randomly picks a bit $\mu \in \{0, 1\}$. If $\mu = 1$, it randomly picks a random session key R in the key space and gives $(R, \mathsf{CT}^*_{\mathbb{A}^*})$ to $\mathcal{A}$. Otherwise, it gives $(\mathsf{Key}^*, \mathsf{CT}^*_{\mathbb{A}^*})$ to $\mathcal{A}$.
- **Phase 2.** $\mathcal{A}$ continues the key queries as in **Phase 1.** under the restriction that none of the queried keys can decrypt $\mathsf{CT}^*_{\mathbb{A}^*}$.
- **Guess:** $\mathcal{A}$ outputs its guess μ'.

Definition 1: PPCMM is adaptively secure if the advantage $\left|\Pr[\,\mu' = \mu\,] - \frac{1}{2}\right|$ of any PPT adversary $\mathcal{A}$ is negligible in the above game.

IV. PRELIMINARIES

A. LINEAR SECRET SHARING SCHEMES (LSSS)

We adopt the definition of LSSS from [18], [28].

Definition 2 (LSSS): Let $\mathcal{U} = (An_1, An_2, \ldots, An_n)$ denote the system attribute universe, where each attribute An_x is composed of two parts: the attribute name An_x and n_x attribute values. $\mathcal{VU}_x = \{\iota_{x,1}, \iota_{x,2}, \ldots, \iota_{x,n_x}\}$ denotes the set of all possible values of An_x.

$\mathbf{A} \in \mathbb{Z}_p^{\ell \times n}$ refers to a share-generating matrix and ρ maps the $i-th$ row of $\mathbf{A}$ to an attribute name $An_x \in \mathcal{U}$. An LSSS consists of the following algorithms:

- **Secret Share**: This algorithm takes in a secret value $s \in \mathbb{Z}_p$ and computes the secret share $\lambda_x = A_x \cdot v$ for each row A_x of $\mathbf{A}$, where $v = (s, y_2, \ldots, y_n)^{\mathrm{T}}$ and $y_2, \ldots, y_n$ are randomly chosen from $\mathbb{Z}_p$.
- **Secret Reconstruction**: This algorithm takes the secret shares $\{\lambda_x\}$ and any authorized set $\mathcal{P}$. It sets $\mathcal{I} = \{i | \rho(i) \in \mathcal{P}\} \subseteq \{1, 2, \ldots, \ell\}$ and calculates the coefficients $\{\omega_i \in \mathbb{Z}_p\}_{i \in \mathcal{I}}$ such that $\sum_{i \in \mathcal{I}} \omega_i A_i = (1, 0, \ldots, 0)$. Then s can be reconstructed by $s = \sum_{i \in \mathcal{I}} \omega_i \lambda_i$.

In our scheme, let $\mathcal{S} = (\mathcal{NAM}_S, \mathcal{VU}_S)$ be the user's attribute set, where $\mathcal{NAM}_S \subseteq \mathbb{Z}_N$ is the attribute name index and $\mathcal{VU}_S = \{J_{x,i}\}_{x \in \mathcal{NAM}_S}$ is the attribute value set. Denote $\mathbb{A} = (\mathbf{A}, \rho, \mathcal{T})$ as the access structure, where $\mathcal{T} = (t_{\rho(1)}, t_{\rho(2)}, \ldots, t_{\rho(\ell)})$ refers to the set of attribute value for each row of $\mathbf{A}$. $\mathcal{S}$ satisfies $\mathbb{A}$ means that there exists $\mathcal{I} \subseteq \{1, 2, \ldots, \ell\}$ satisfying $(\mathbf{A}, \rho)$, $\{\rho(i) | i \in \mathcal{I}\} \subseteq \mathcal{NAM}_S$ and $J_{\rho(i)} = t_{\rho(i)}\ \forall i \in \mathcal{I}$.

B. COMPOSITE ORDER BILINEAR GROUPS

The adopted composite order bilinear groups are defined as in [4], [18]. By taking in a security parameter λ, a group generator $\mathcal{G}$ outputs the terms $(\mathbb{G}, \mathbb{G}_1, p_1, p_2, p_3, p_4, e)$. The order of cyclic groups $\mathbb{G}$ and $\mathbb{G}_1$ is $N = p_1 p_2 p_3 p_4$, where p_1, p_2, p_3 and p_4 are 4 distinct primes. $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ is a bilinear map with such properties:

1. Bilinearity: $\forall \varrho, \varpi \in \mathbb{G}$ and $i, j \in \mathbb{Z}_N$, we have $e(\varrho^i, \varpi^j) = e(\varrho, \varpi)^{ij}$.

2. Non-degeneracy:$\exists\ \varrho \in \mathbb{G}$ such that $e(\varrho, \varrho)$ has order N in $\mathbb{G}_1$.

Let $\mathbb{G}_{p_x}$ be the subgroup of order p_x in $\mathbb{G}$. If $\varrho_x \in \mathbb{G}_{p_x}$ and $\varrho_y \in \mathbb{G}_{p_y}$, for $x \neq y$, we have $e(\varrho_x, \varrho_y) = 1$.

C. COMPLEXITY ASSUMPTION

Let $\mathbb{GG}$ be the terms $(\mathbb{G}, \mathbb{G}_1, N = p_1 p_2 p_3 p_4, e)$ generated by $\mathcal{G}$.

Assumption 1. Given $\mathcal{G}$ and the following distribution:

$$g \xleftarrow{R} \mathbb{G}_{p_1}, E_3 \xleftarrow{R} \mathbb{G}_{p_3}, E_4 \xleftarrow{R} \mathbb{G}_{p_4},$$
$$\Phi = (\mathbb{GG}, g, E_3, E_4), \eth_1 \xleftarrow{R} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2}, \eth_2 \xleftarrow{R} \mathbb{G}_{p_1}.$$

The algorithm $\mathcal{A}$'s advantage in breaking this assumption is $\mathsf{Adv1}_{\mathcal{G},\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(\Phi, \eth_1) = 1] - \Pr[\mathcal{A}(\Phi, \eth_2) = 1]|$.

Definition 3: $\mathcal{G}$ satisfies Assumption 1 if $\mathsf{Adv1}_{\mathcal{G},\mathcal{A}}(\lambda)$ is negligible for any probabilistic polynomial time (PPT) algorithm $\mathcal{A}$.

Assumption 2. Given $\mathcal{G}$ and the following distribution:

$$g, E_1 \xleftarrow{R} \mathbb{G}_{p_1}, E_2, G_2 \xleftarrow{R} \mathbb{G}_{p_2}, E_3, G_3 \xleftarrow{R} \mathbb{G}_{p_3}, E_4 \xleftarrow{R} \mathbb{G}_{p_4},$$
$$\Phi = (\mathbb{GG}, g, E_1 E_2, G_2 G_3, E_3, E_4),$$
$$\eth_1 \xleftarrow{R} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}, \eth_2 \xleftarrow{R} \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}.$$

The algorithm $\mathcal{A}$'s advantage in breaking this assumption is $\mathsf{Adv2}_{\mathcal{G},\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(\Phi, \eth_1) = 1] - \Pr[\mathcal{A}(\Phi, \eth_2) = 1]|$.

Definition 4: $\mathcal{G}$ satisfies Assumption 2 if $\mathsf{Adv2}_{\mathcal{G},\mathcal{A}}(\lambda)$ is negligible for any PPT algorithm $\mathcal{A}$.

Assumption 3. Given $\mathcal{G}$ and the following distribution:

$$g \xleftarrow{R} \mathbb{G}_{p_1}, g_2, E_2, G_2 \xleftarrow{R} \mathbb{G}_{p_2}, E_3 \xleftarrow{R} \mathbb{G}_{p_3}, E_4 \xleftarrow{R} \mathbb{G}_{p_4},$$
$$\Phi = (\mathbb{GG}, g, g_2, g^{\alpha} E_2, g^{s} G_2, E_3, E_4),$$
$$\eth_1 = \hat{e}(g, g)^{\alpha s}, \eth_2 \xleftarrow{R} \mathbb{G}_1.$$

The algorithm $\mathcal{A}$'s advantage in breaking this assumption is $\mathsf{Adv3}_{\mathcal{G},\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(\Phi, \eth_1) = 1] - \Pr[\mathcal{A}(\Phi, \eth_2) = 1]|$.

Definition 5: $\mathcal{G}$ satisfies Assumption 3 if $\mathsf{Adv3}_{\mathcal{G},\mathcal{A}}(\lambda)$ is negligible for any PPT algorithm $\mathcal{A}$.

Assumption 4. Given $\mathcal{G}$ and the following distribution:

$$g, h \xleftarrow{R} \mathbb{G}_{p_1}, g_2, E_2, A_2, B_2, G_2 \xleftarrow{R} \mathbb{G}_{p_2}, t', r' \xleftarrow{R} \mathbb{Z}_N$$
$$E_3 \xleftarrow{R} \mathbb{G}_{p_3}, E_4, Z, A_4, G_4 \xleftarrow{R} \mathbb{G}_{p_4},$$
$$\Phi = (\mathbb{GG}, g, g_2, g^{t'} B_2, h^{t'} G_2, E_3, E_4, hZ, g^{r'} G_2 G_4),$$
$$\eth_1 = h^{r'} A_2 A_4, \eth_2 \xleftarrow{R} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_4}.$$

The algorithm $\mathcal{A}$'s advantage in breaking this assumption is $\mathsf{Adv4}_{\mathcal{G},\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(\Phi, \eth_1) = 1] - \Pr[\mathcal{A}(\Phi, \eth_2) = 1]|$.

Definition 6: $\mathcal{G}$ satisfies Assumption 4 if $\mathsf{Adv4}_{\mathcal{G},\mathcal{A}}(\lambda)$ is negligible any PPT algorithm $\mathcal{A}$.

TABLE 1. Notations employed in PPCMM.

Notation	Description
λ	The system security parameter
AA	Attribute authority
MDO	Multimedia data owner
MDU	Multimedia data user
SP	System public parameters
MSK	Master secret key
$x \in_R X$	x is randomly chosen from the set X
$\mathbb{G}_{p_i}$	A subset of group $\mathbb{G}$ of order p_i
$\mathbb{A}$	Access policy
$\mathbf{A}$	an $\ell \times n$ LSSS matrix ($\ell < \jmath$)
ρ	ρ maps each row A_x of $\mathbf{A}$ to an attribute name
$\mathcal{T}$	the concrete value set of attributes involved in $\mathbb{A}$
IC	Intermediate ciphertext
IC_{MA}	Intermediate ciphertext for matching test
IC_{DE}	Intermediate ciphertext for some access policies
$\mathsf{CT}_{\mathbb{A}}$	The final ciphertext of $\mathbb{A}$
CT_{MA}	Final ciphertext for matching test of specific $\mathbb{A}$
CT_{DE}	Final ciphertext for specific $\mathbb{A}$
SK_u	The intermediate key of user u
UDK	User decryption key of user u
DEK	Data encryption key to encrypt multimedia data
PDC	Partially decrypted ciphertext of DEK

V. DESIGN DETAILS OF PPCMM

This section gives the detailed construction of PPCMM. Table 1 presents the description of notations used in PPCMM.

1) SYSTEM INITIALIZATION

AA obtains the terms $\mathbb{GG} = (e, \mathbb{G}, \mathbb{G}_1, N, p_1, p_2, p_3, p_4)$ by taking in a security parameter λ. The system attribute universe is set as $U = \mathbb{Z}_N$. It then runs the setup algorithm as follows.

- **System Setup**: AA uniformly selects $g, f \in_R \mathbb{G}_{p_1}$, $\alpha, \beta \in_R \mathbb{Z}_N$, $\Im_3 \in_R \mathbb{G}_{p_3}$, $B, \Im_4 \in_R \mathbb{G}_{p_4}$ and computes $\wp = e(g,g)^{\alpha}$, $F = fB$. It publishes $\mathsf{SP} = (\wp, g, g^{\beta}, N, F, \Im_4)$ and keeps $\mathsf{MSK} = (\alpha, f, \Im_3)$ as secret.

2) MDU AUTHORIZATION

When MDU joins in the system, he is issued a set of attributes $\mathcal{S} = (\mathcal{NAMS}, \mathcal{VUS})$, where $\mathcal{NAMS} \subseteq \mathbb{Z}_N$ and $\mathcal{VUS} = \{s_i\}_{i \in \mathcal{NAMS}}$. AA sets the access right by running the **KeyGen** algorithm.

- **KeyGen**: It selects $t, u \in_R \mathbb{Z}_N$, $Q, Q', Q_i \in_R \mathbb{G}_{p_3}$ for $i \in \mathcal{NAMS}$ and computes $K = g^{\alpha u} g^{\beta t} Q$, $L = g^t Q'$ and $K_i = (g^{s_i} f)^t Q_i$. It then sets the user decryption key $UDK = u$. It finally gives UDK and the intermediate key $SK_u = (\mathcal{S}, K, L, \{K_i\}_{i \in \mathcal{I}_S})$ to the user.

3) MOBILE MULTIMEDIA DATA OUTSOURCING

By running the **Offline Encryption** algorithm, MDO could accomplish the pre-computation procedure and create an intermediate ciphertext. After knowing the explicit multimedia data and the specified access policy $\mathbb{A} = (\mathbf{A}, \rho, \mathcal{T})$, MDO then runs the **Online Encryption** algorithm to generate the final ciphertext.

- **Offline Encryption**: Here we assume that the number of rows employed in an LSSS access policy can not overrun a maximum bound $\jmath$. This algorithm first selects $s, s_1 \in_R \mathbb{Z}_N$ and $B_{MA} \in_R \mathbb{G}_{p_4}$. For $j = 1$ to $\jmath$, it selects $\lambda'_{MA,j}, x_{MA,j}, \lambda'_j, x_j, r_j \in_R \mathbb{Z}_N$, $B_{MA}, B_{MA,j}, B_{c,j}, B_{d,j} \in_R \mathbb{G}_{p_4}$ and computes:
$MAK = \wp^{s_1}$, $C_{MA} = g^{s_1} B_{MA}$, $C_{MA,j} = g^{\beta \lambda'_{MA,j}} (g^{x_{MA,j}} F)^{-s_1} B_{MA,j}$
$DEK = \wp^{s}$, $C_0 = g^{s}$, $C_j = g^{\beta \lambda'_j} (g^{x_j} F)^{-r_j} B_{c,j}$, $D_j = g^{r_j} B_{d,j}$.
The intermediate ciphertext is set as $IC = (IC_{MA}, IC_{DE})$, where $IC_{MA} = (MAK, C_{MA}, s_1, \{C_{MA,j}, \lambda'_{MA,j}, x_{MA,j}\}_{j \in [1,\jmath]})$ and $IC_{DE} = (DEK, C_0, s, \{C_j, D_j, \lambda'_j, x_j, r_j\}_{j \in [1,\jmath]})$.
- **Online Encryption**: Given IC and a specified access policy $\mathbb{A} = (\mathbf{A}, \rho, \mathcal{T})$, where $\mathbf{A}$ refers to an $\ell \times n$ LSSS matrix ($\ell < \jmath$), ρ maps each row A_x of $\mathbf{A}$ to an attribute name, and $\mathcal{T} = (t_{\rho(1)}, t_{\rho(2)}, \ldots, t_{\rho(\ell)}) \in \mathbb{Z}_N^{\ell}$. This algorithm selects $y_2, \ldots, y_n, v_2, \ldots, v_n \in_R \mathbb{Z}_N$. It then computes $\overrightarrow{\lambda}_{MA} = (\lambda_{MA,1}, \lambda_{MA,2}, \ldots, \lambda_{MA,\ell})^T = \mathbf{A}(s_1, y_2, \ldots, y_n)^T$ and $\overrightarrow{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_{\ell})^T = \mathbf{A}(s, v_2, \ldots, v_n)^T$.
For $j = 1$ to ℓ, it calculates $D_{MA,j} = \lambda_{MA,j} - \lambda'_{MA,j}$, $E_{MA,j} = s_1(x_{MA,j} - t_{\rho(j)})$,
$C_{1,j} = \lambda_j - \lambda'_j$, $D_{1,j} = r_j(x_j - t_{\rho(j)})$.
The final ciphertext is set as $\mathsf{CT}_{\mathbb{A}} = ((\mathbf{A}, \rho), CT_{MA}, CT_{DE})$, where $CT_{MA} = (MAK, C_{MA}, \{C_{MA,j}, D_{MA,j}, E_{MA,j}\}_{j \in [1,\ell]})$ and $CT_{DE} = (C_0, \{C_j, D_j, C_{1,j}, D_{1,j}\}_{j \in [1,\ell]})$.
DEK is set as the session key **Key**. MDO then employs DEK as the symmetric data encryption key to encrypt his multimedia data M. Finally, $\mathsf{CT}_{\mathbb{A}}$ and the encrypted data M_{DEK} will be uploaded to CSP.

4) DATA ACCESS

MDU can retrieve and obtain the ciphertext tuple $(M_{DEK}, \mathsf{CT}_{\mathbb{A}})$. If the attribute set matches the access policy in $\mathsf{CT}_{\mathbb{A}}$, DEK can be recovered by the following algorithms.

- **Ciphertext Transmission**: Given $\mathsf{CT}_{\mathbb{A}}$ in the form of $((\mathbf{A}, \rho), CT_{MA}, CT_{DE})$, MDU replaces MAK by $MAK' = MAK^u$ and calls CSP to provide outsourced decryption service for the new $\mathsf{CT}'_{\mathbb{A}}$. MDU also sets $SK'_u = (\mathcal{NAMS}, K, L, \{K_i\}_{i \in \mathcal{I}_S})$.
- **Matching Test**: Given $(\mathbf{A}, \rho)$, SK'_u and $CT'_{MA} = (MAK', C_{MA}, \{C_{MA,j}, D_{MA,j}, E_{MA,j}\}_{j \in [1,\ell]})$, CSP first sets $\mathbf{I}_{\mathbf{A},\rho}$ which refers to the set of minimum subsets of $\{1, 2, \ldots, \ell\}$ that satisfies $(\mathbf{A}, \rho)$. It then checks if there is a subset $\mathcal{I} \in \mathbf{I}_{\mathbf{A},\rho}$ satisfying $\{\rho(i) | i \in \mathcal{I}\} \subseteq \mathcal{NAMS}$ and

$$MAK'^{-1} = e\left(\prod_{i \in \mathcal{I}} (C_{MA,i} \cdot (g^{\beta})^{D_{MA,i}} \cdot g^{E_{MA,i}})^{\omega_i}, L\right) \cdot e\left(C_{MA}, K^{-1} \prod_{i \in \mathcal{I}} K_{\rho(i)}^{\omega_i}\right)$$

where $\sum_{i \in \mathcal{I}} \omega_i A_i = (1, 0, \ldots, 0)$ for some constants $\{\omega_i\}_{i \in \mathcal{I}}$. If no such subset $\mathcal{I}$ exists, this algorithm

outputs $\perp$. Otherwise, CSP runs the **Partial Decryption** algorithm with the eligible $\{\omega_i\}$ and $\mathcal{I}$.

- **Partial Decryption**: Given $\mathcal{I}$, $\{\omega_i\}$, CT_{DE} and SK'_u, CSP computes

$$PDC = \frac{e(C_0, K)}{\prod_{i\in\mathcal{I}} \left(e(C_i \cdot (g^{\beta})^{C_{1,i}} \cdot g^{D_{1,i}}, L)e(D_i, K_{\rho(i)})\right)^{\omega_i}} = e(g,g)^{\alpha us}$$

and gives PDC to MDU.

- **User Decryption**: MDU recovers DEK by $DEK = (PDC)^{1/UDK}$ and further decrypts M_{DEK}.

A. DISCUSSION

Note that, MDO in the propose scheme is assumed to prepare an IC with at most $\jmath$ attribute components. Actually, One may adopt the technique in [29] to remove such restriction. In addition, MDO can prepare multiple ICs in a intermediate pooling when the battery is in charging. It can also offload the preparatory computation task to other available devices.

VI. SECURITY ANALYSIS

For simplicity, we reduce the security of PPCMM to the underline ZZD scheme [18]. We denote PPCMM and ZZD scheme as $\sum_{PPCMM}$ and $\sum_{ZZD}$, respectively.

Theorem 1: Assume that $\sum_{ZZD}$ is adaptively secure, our proposed $\sum_{PPCMM}$ is adaptively secure against the chosen plaintext attack in the security game which is defined in Section III-C.

Proof:

Suppose that there exists an adversary $\mathcal{A}$ that can break $\sum_{PPCMM}$, then we can construct a simulator $\mathcal{B}$ to break $\sum_{ZZD}$.

- **Setup.** $\mathcal{B}$ obtains the public parameters SP from $\sum_{ZZD}$ and gives it to $\mathcal{A}$.
- **Phase 1.** $\mathcal{B}$ initializes an integer counter $\hbar = 0$, an empty table T and an empty set D. It then answers the key queries from $\mathcal{A}$ as follows:
  - $\mathsf{Create}(\mathcal{S})$: After receiving the requested set $\mathcal{S}$ from $\mathcal{A}$, $\mathcal{B}$ sets $\hbar := \hbar + 1$ and gives $\mathcal{S}$ to $\sum_{ZZD}$, which then returns a key in the form of $SK_{\mathcal{S}} = (\mathcal{S}, K, L, \{K_i\}_{i\in\mathcal{NAMS}})$, where
    $\overline{K} = g^{\alpha} g^{\beta\bar{t}} \overline{Q}$,
    $\overline{L} = g^{\bar{t}} \overline{Q}'$
    $\forall i \in \mathcal{NAMS}$ $\overline{K}_i = (g^{s_i} f)^{\bar{t}} \overline{Q}_i$.
    $\mathcal{B}$ randomly selects $u \in \mathbb{Z}_N$ and computes
    $K = (\overline{K})^u = g^{\alpha u} g^{\beta \bar{t} u} \overline{Q}^u$,
    $L = (\overline{L})^u = g^{\bar{t}u} \overline{Q}'^u$
    $\forall i \in \mathcal{NAMS}$ $K_i = (\overline{K}_i)^u = (g^{s_i} f)^{\bar{t}u} \overline{Q}_i^u$.
    It is implied that $t = \bar{t}u$, $Q = \overline{Q}^u$, $Q, = \overline{Q}'^u$ and $Q_i = \overline{Q}_i^u$.
    $\mathcal{B}$ sets $\mathsf{UDK} = u$ and then stores $(\hbar, \mathcal{S}, \mathsf{UDK}, \mathsf{SK}_{\mathcal{S}})$ in T.
  - $\mathsf{Corrupt}(i)$: $\mathcal{B}$ checks if the i-th entry $(\hbar, \mathcal{S}, \mathsf{UDK}, \mathsf{SK}_{\mathcal{S}})$ can be found in T. If so, it sets $D := D \bigcup \{\mathcal{S}\}$ and gives $(\mathsf{UDK}, \mathsf{SK}_{\mathcal{S}})$ to $\mathcal{A}$. Otherwise, it outputs $\perp$.
- **Challenge.** After receiving the submitted challenge $\mathbb{A}^*$ from $\mathcal{A}$, $\mathcal{B}$ randomly selects two equal length message M_0, M_1 and sends them to $\sum_{ZZD}$ along with $\mathbb{A}^*$. $\sum_{ZZD}$ returns a ciphertext in the form of
  $CC = (CC_{MA}, CC_{DE})$, where $CC_{MA} = (\overline{MAK} = \wp^{s_1}, \overline{C}_{MA} = g^{s_1}\overline{B}_{MA}, \{\overline{C}_{MA,j} = g^{\beta\lambda_{MA,j}}(g^{t_{\rho(j)}}F)^{-s_1}\overline{B}_{MA,j}\}_{j\in[1,\ell]})$ and
  $CC_{DE} = (\overline{C} = M_v e(g,g)^{\alpha s}, \overline{C}_0 = g^s, \{\overline{C}_j = g^{\beta\lambda_j}(g^{x_j}F)^{-r_j}\overline{B}_{c,j}, \overline{D}_j = g^{r_j}\overline{B}_{d,j}\}_{j\in[1,\ell]})$.
  $\mathcal{B}$ randomly selects $\lambda'_{MA,j}, x_{MA,j}, \lambda'_j, x_j \in_R \mathbb{Z}_N$ and computes
  $C_{MA,j} = \overline{C}_{MA,j} \cdot g^{-a\lambda'_{MA,j}} \cdot g^{-x_{MA,j}} = g^{\beta\lambda_{MA,j}}(g^{t_{\rho(j)}}F)^{-s_1}\overline{B}_{MA,j} \cdot g^{-\beta\lambda'_{MA,j}} \cdot g^{-x_{MA,j}}$.
  $C_j = \overline{C}_j \cdot g^{-\beta\lambda'_j} \cdot g^{-x_j} = g^{\beta\lambda_j}(g^{x_j}F)^{-r_j}\overline{B}_{c,j} \cdot g^{-\beta\lambda'_j} \cdot g^{-x_j}$
  $\mathcal{B}$ sets $MAK = \overline{MAK}$, $C_{MA} = \overline{C}_{MA}$, $D_{MA,j} = \lambda'_{MA,j}$, $E_{MA,j} = x_{MA,j}$, $C_0 = \overline{C}_0$, $D_j = \overline{D}_j$, $C_{1,j} = \lambda'_j$, $D_{1,j} = x_j$.
  $\mathcal{B}$ guesses which message $v_{\mathcal{B}} \in \{0,1\}$ is encrypted and computes $\mathsf{Key}_{guess} = \frac{\overline{C}}{M_{v_{\mathcal{B}}}}$. It finally gives $(\mathsf{Key}_{guess}, \mathsf{CT}^*_{\mathbb{A}^*} = (CT_{MA}, CT_{DE}))$ to $\mathcal{A}$, where $CT_{MA} = (MAK, C_{MA}, \{C_{MA,j}, D_{MA,j}, E_{MA,j}\}_{j\in[1,\ell]})$ and $CT_{DE} = (C_0, \{C_j, D_j, C_{1,j}, D_{1,j}\}_{j\in[1,\ell]})$.
- **Phase 2.** $\mathcal{B}$ acts the same as in **Phase 1.** under the restriction that none of the queried keys can decrypt $\mathsf{CT}^*_{\mathbb{A}^*}$.
- **Guess:** $\mathcal{A}$ outputs its guess μ'.

■

VII. PERFORMANCE COMPARISON

In this section, we first give the characteristic comparison between PPCMM and some related works [5], [18], [27]–[29], [33]–[36]. Then, we give the numeric and experimental results of the adaptively secure schemes [18], [28] and ours.

A. CHARACTERISTIC COMPARISON

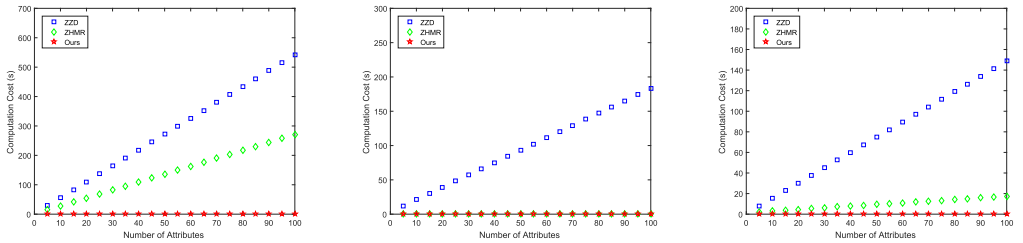
Table 2 provides the comparison of important characteristics including large universe, adaptive security, policy expressiveness, group form, hidden policy, online/offline encryption and outsourced decryption. As shown in Table 2, only the scheme [18], [27], [28] and ours achieve the adaptive security. The scale of system attribute universe in the schemes [27], [34], [35] is not large universe. Among the schemes [18], [27], [28], [34]–[36] with hidden policy and ours, only the scheme [36] and ours can support online/offline encryption technique which could save a grate deal of computation time in embedding the specific access policy. The scheme in [33] can securely outsource most of the decryption overhead to the cloud. However, the access policy in [33] is in the plaintext form and the offline encryption is not addressed. In summary, PPCMM is the only one scheme which can simultaneously support the above important features.

TABLE 2. Characteristic comparison with related work.

Schemes	Large Universe	Adaptive Security	Expressiveness	Group	Policy Hidden	Online/Offline	Outsourced Decryption
[5]	✓	×	LSSS	Prime	×	×	×
[34]	×	×	AND Gate	Prime	✓	×	×
[27]	×	✓	AND Gate	Composite	✓	×	×
[18]	✓	✓	LSSS	Composite	✓	×	×
[35]	×	×	ABF	Prime	✓	×	×
[33]	✓	×	LSSS	Prime	×	×	✓
[29]	✓	×	LSSS	Prime	×	✓	×
[36]	✓	×	ABF	Prime	✓	✓	×
[28]	✓	✓	LSSS	Composite	✓	×	×
Ours	✓	✓	LSSS	Composite	✓	✓	✓

TABLE 3. Computation cost comparison.

Schemes	Key Generation	Encryption		Matching Test		Decryption	
		Offline	Online	Cloud	User	Cloud	User
[18]	$(2\|S_K\|+3)E_1$	✗	$(6\|S_C\|+2)E_1+2E_2$	✗	$(2\|I\|+2)E_1+2P$	✗	$\|I\|E_2+(2\|I\|+1)P$
[28]	$(2\|S_K\|+4)E_1$	✗	$(3\|S_C\|+1)E_1+1E_2$	✗	✗	✗	$2\|I\|E_1+2P$
Ours	$(2\|S_K\|+3)E_1$	$(6\jmath+2)E_1+2E_2$	✗	$(4\|I\|+2)E_1+2P$	✗	$\|I\|(2E_1+E_2)+(2\|I\|+1)P$	$1E_2$



(a) Online Encryption (b) User Matching Test (c) User Decryption

FIGURE 2. Time of online encryption and data access.

B. NUMERIC COMPARISON

Table 3 compares the schemes [18], [28] with PPCMM in terms of the numeric computation cost in the stage of key generation, encryption, matching test and decryption. The computation cost is calculated according to the time of employed modular exponential operation and bilinear pairing operation. Let E_1, E_2 and P be a modular exponential operation in $\mathbb{G}$, a modular exponential operation in $\mathbb{G}_1$ and a bilinear pairing operation, respectively. S_K, S_C and I represent to the attribute set involved in the stage of key generation, encryption and decryption, respectively. From Table 3, we find that the schemes [18], [28] and ours spend almost the same computation overhead in generating the keys for users. Thanks to the online/offline encryption technique, we could pre-compute the intermediate ciphertext for at most $\jmath$ attributes during the offline stage. To generate the final ciphertext for a specific access policy, MDO needs to execute $4\jmath$ times of modular integer operation in group $\mathbb{Z}_N$ which costs much less computation resource than the complicated modular exponential operation or bilinear pairing operation. Although the scheme [28] do not need to check if the user's attributes exactly match the partially hidden access policy before decryption, it remains the risk that the decrypted ciphertext might not be the correct plaintext message because that the constants $\{\omega_i\}$ are computed assuming the attribute set matches the attribute names in the access policy rather than the hidden attribute values. In contrast, the scheme [18] and ours utilize the matching test technique to check if the set $\mathcal{S}$ precisely satisfies the access policy including both the attribute names and the hidden attribute values. Such matching test technique may significantly avoid the sequential useless computation overhead if only the attribute names match the access policy. In PPCMM, since that most of the computation overhead in the stage of matching test and decryption is offloaded to CSP, only one modular exponential operation in $\mathbb{G}_1$ is required on the user side. Due to the effective online/offline encryption technique and outsourced decryption method, PPCMM achieves both efficient online encryption and constant size of user decryption overhead.

C. EXPERIMENT RESULT

We implement PPCMM, ZZD scheme [18] and ZHMR scheme [28] on a windows laptop with 2.90 GHz Intel Pentium(R) CPU by adopting Type A1 pairings from the Java Pairing Based Cryptography (JPBC) [37]. We mainly count the time of pairing and exponennt operations. Fig. 2 gives the implementation results in terms of the time of online encryption, user matching test and user decryption. The implementation results indicate that PPCMM significantly saves the

computation time in the phase of online encryption, matching test and user decryption.

VIII. CONCLUSION

In this work, we designed PPCMM, a privacy-preserving cloud-assisted mobile multimedia data access control scheme, which simultaneously addressed the policy privacy and the efficiency of both online encryption and user decryption. PPCMM can also support large universe and any LSSS monotonic access policy. In the access policy, the sensitive attribute values are hidden in the final ciphertext and only the generic attribute names are clear. In the offline stage, MDO can pre-compute the intermediate ciphertext elements for at most $\jmath$ random attributes locally or on a third party. Then in the online stage, the final ciphertext can be rapidly accomplished by combining the intermediate ciphertext with a specific access policy for certain multimedia data. Due to the effective decryption outsourcing approach, the computation cost of a user to realize the matching test and final decryption is reduced to only one exponent operation, no matter how many attributes are involved. The security analysis and performance comparison demonstrated that PPCMM is secure, efficient and practical.

Although the works [33], [38] has introduced the verifiability mechanism for decryption outsourced ABE with entirely public access policy. However, their method can not be directly adopted in the ABE with a partial hidden policy. We leave this issue as our future work.

REFERENCES

[1] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology— EUROCRYPT*, R. Cramer, Ed. Berlin, Germany: Springer, 2005, pp. 457–473.

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

[3] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, New York, NY, USA, 2007, pp. 456–465.

[4] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Advances in Cryptology—EUROCRYPT*, H. Gilbert, Ed. Berlin, Germany: Springer, 2010, pp. 62–91.

[5] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, Nov. 2013, pp. 463–474.

[6] D. Wu, J. Yan, H. Wang, D. Wu, and R. Wang, "Social attribute aware incentive mechanism for device-to-device video distribution," *IEEE Trans. Multimedia*, vol. 19, no. 8, pp. 1908–1920, Aug. 2017.

[7] C. Luo, J. Ji, Q. Wang, X. Chen, and P. Li, "Channel state information prediction for 5G wireless communications: A deep learning approach," *IEEE Trans. Netw. Sci. Eng.*, to be published.

[8] D. Wu, Q. Liu, H. Wang, D. Wu, and R. Wang, "Socially aware energy-efficient mobile edge collaboration for video distribution," *IEEE Trans. Multimedia*, vol. 19, no. 10, pp. 2197–2209, Oct. 2017.

[9] Z. Zhang, C. Wang, C. Gan, S. Sun, and M. Wang, "Automatic modulation classification using convolutional neural network with features fusion of SPWVD and BJD," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 3, pp. 469–478, Sep. 2019.

[10] D. Wu, L. Deng, H. Wang, K. Liu, and R. Wang, "Similarity aware safety multimedia data transmission mechanism for Internet of vehicles," *Future Gener. Comput. Syst.*, vol. 99, pp. 609–623, Oct. 2019.

[11] J. Xiong, J. Ren, L. Chen, Z. Yao, M. Lin, D. Wu, and B. Niu, "Enhancing privacy and availability for data clustering in intelligent electrical service of IoT," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1530–1540, Apr. 2018.

[12] D. Wu, H. Shi, H. Wang, R. Wang, and H. Fang, "A feature-based learning system for Internet of Things applications," *IEEE Internet Things J.*, vol. IOT-6, no. 2, pp. 1928–1937, Apr. 2019.

[13] L. Wang, Y. Tian, D. Zhang, and Y. Lu, "Constant-round authenticated and dynamic group key agreement protocol for D2D group communications," *Inf. Sci.*, vol. 503, pp. 61–71, Nov. 2019.

[14] Y. Zhang, R. Deng, D. Zheng, J. Li, P. Wu, and J. Cao, "Efficient and robust certificateless signature for data crowdsensing in cloud-assisted industrial IoT," *IEEE Trans. Ind. Informat.*, to be published.

[15] J. Xiong, Y. Zhang, L. Lin, J. Shen, X. Li, and M. Lin, "ms-PoSW: A multi-server aided proof of shared ownership scheme for secure deduplication in cloud," *Concurrency Comput., Pract. Exp.*, p. e4252, 2017. doi: 10.1002/cpe.4252.

[16] H. Wang, D. He, and J. Han, "VOD-ADAC: Anonymous distributed fine-grained access control protocol with verifiable outsourced decryption in public cloud," *IEEE Trans. Services Comput.*, to be published.

[17] Y. Zhang, R. Deng, X. Liu, and D. Zheng, "Outsourcing service fair payment based on blockchain and its applications in cloud computing," *IEEE Trans. Services Comput.*, to be published.

[18] Y. Zhang, D. Zheng, and R. H. Deng, "Security and privacy in smart health: Efficient policy-hiding attribute-based access control," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2130–2145, Jun. 2018.

[19] 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.*, New York, NY, USA, Oct. 2006, pp. 89–98.

[20] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proc. IEEE INFOCOM*, San Diego, CA, USA, Mar. 2010, pp. 1–9.

[21] J. Li, Z. Guan, X. Du, Z. Zhang, and J. Wu, "An efficient encryption scheme with verifiable outsourced decryption in mobile cloud computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[22] Y. Yang, X. Liu, and R. H. Deng, "Lightweight break-glass access control system for healthcare Internet-of-Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3610–3617, Aug. 2017.

[23] Y. Yang, X. Liu, R. H. Deng, and Y. Li, "Lightweight sharable and traceable secure mobile health system," *IEEE Trans. Dependable Secure Comput.*, to be published.

[24] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *Applied Cryptography and Network Security* , S. M. Bellovin, R. Gennaro, A. Keromytis, and M. Yung, Eds. Berlin, Germany: Springer, 2008, pp. 111–129.

[25] J. Lai, R. H. Deng, and Y. Li, "Fully secure cipertext-policy hiding cp-abe," in *Information Security Practice and Experience*, F. Bao and J. Weng, Eds. Berlin, Germany: Springer, 2011, pp. 24–39.

[26] C. Jin, X. Feng, and Q. Shen, "Fully secure hidden ciphertext policy attribute-based encryption with short ciphertext size," in *Proc. 6th Int. Conf. Commun. Netw. Secur.*, New York, NY, USA, Nov. 2016, pp. 91–98.

[27] J. Lai, R. H. Deng, and Y. Li, "Expressive CP-ABE with partially hidden access structures," in *Proc. 7th ACM Symp. Inf., Comput. Commun. Secur.*, New York, NY, USA, May 2012, pp. 18–19.

[28] L. Zhang, G. Hu, Y. Mu, and F. Rezaeibagha, "Hidden ciphertext policy attribute-based encryption with fast decryption for personal health record system," *IEEE Access*, vol. 7, pp. 33202–33213, 2019.

[29] S. Hohenberger and B. Waters, "Online/offline attribute-based encryption," in *Public-Key Cryptography—PKC*, H. Krawczyk, Ed. Berlin, Germany: Springer, 2014, pp. 293–310.

[30] 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, Nov. 2016.

[31] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in *Proc. 20th USENIX Conf. Secur.*, Berkeley, CA, USA, Aug. 2011, p. 34.

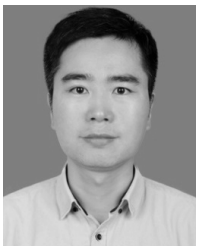
[32] Q. Li, J. Ma, R. Li, X. Liu, J. Xiong, and D. Chen, "Secure, efficient and revocable multi-authority access control system in cloud storage," *Comput. Secur.*, vol. 59, pp. 45–59, Jun. 2016.

[33] J. Ning, Z. Cao, X. Dong, H. Ma, L. Wei, and K. Liang, "Auditable σ-time outsourced attribute-based encryption for access control in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 1, pp. 94–105, Jan. 2018.

[34] T. V. X. Phuong, G. Yang, and W. Susilo, "Hidden ciphertext policy attribute-based encryption under standard assumptions," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 1, pp. 35–45, Jun. 2016.
[35] K. Yang, Q. Han, H. Li, K. Zheng, Z. Su, and X. Shen, "An efficient and fine-grained big data access control scheme with privacy-preserving policy," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 563–571, Apr. 2017.
[36] D. Zheng, A. Wu, Y. Zhang, and Q. Zhao, "Efficient and privacy-preserving medical data sharing in Internet of things with limited computing power," *IEEE Access*, vol. 6, pp. 28019–28027, 2018.
[37] A. De Caro and V. Iovino, "jPBC: Java pairing based cryptography," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Kerkyra, Greece, Jun./Jul. 2011, pp. 850–855.
[38] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 8, pp. 1343–1354, Aug. 2013.

QI LI received the Ph.D. degree in computer system architecture from Xidian University, Xi'an, China, in 2014. He is currently a Lecturer with the School of Computer Science, Nanjing University of Posts and Telecommunications, China. He is also a Visiting Researcher with the State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University. His research interests include cloud security, information security, and applied cryptography.

YOULIANG TIAN received the B.Sc. degree in mathematics and applied mathematics, in 2004, the M.Sc. degree in applied mathematics from Guizhou University, in 2009, and the Ph.D. degree in cryptography from Xidian University, in 2012. From 2012 to 2015, he was a Postdoctoral Associate with the State Key Laboratory for Chinese Academy of Sciences. He is currently a Professor and a Ph.D. Supervisor with the College of Computer Science and Technology, Guizhou University. His research interests include algorithm game theory, cryptography, and security protocol.



YINGHUI ZHANG has been a Professor with the School of Cyberspace Security, Xi'an University of Posts and Telecommunications, since 2018. He has published over 80 research articles in ACM ASIACCS, the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, the IEEE TRANSACTIONS ON SERVICES COMPUTING, *Computer Networks*, the IEEE INTERNET OF THINGS JOURNAL, *Computers & Security*, and the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS. His research interests include public key cryptography, cloud security, and wireless network security.

LIMIN SHEN received the B.S. and M.S. degrees in mathematics from Wuhan University, China, in 2001 and 2004, respectively, and the Ph.D. degree in computer systems organization from Xidian University, China, in 2018. She is currently a Lecturer with the School of Computer Science and Technology, Nanjing Normal University. Her research interests mainly include cryptography, information security, and wireless sensor networks.

JINGJING GUO received the M.Sc. and Ph.D. degrees in computer science from Xidian University, Xi'an, China, in 2012 and 2015, respectively, where she is currently a Lecturer with the School of Cyber Engineering. Her research interests include trust management, social networks, access control, and information security.