Soft Computing (2021) 25:10543–10559
https://doi.org/10.1007/s00500-021-05996-8

FOUNDATIONS

Partial policy hiding attribute-based encryption in vehicular fog computing

Tingyun Gan[1] · Yongjian Liao[1] · Yikuan Liang[1] · Zijun Zhou[1] · Ganglin Zhang[1]

Accepted: 20 June 2021 / Published online: 6 July 2021

Abstract
Vehicular fog computing (VFC), combining the vehicular ad hoc network with fog computing, is an efficient vehicle communication architecture. However, the user data is often threatened since VFC is an open environment. Attribute-based encryption (ABE) is suitable for open scenarios, such as cloud and Internet of Things, because of its confidentiality and access control characteristics. However, the traditional ABE has disadvantages, such as the inability to hide the attributes in the access policy and the use of computationally inefficient composite order bilinear pairing groups to prove adaptive security. Traditional ABE is not practical in VFC. We summarized the existing schemes of full policy hiding ABE and partial policy hiding ABE and then concluded that partial policy hiding ABE is more suitable for VFC. We combine policy hiding technology and the technology of converting bilinear pairing cryptography schemes into prime-order bilinear pairing cryptography schemes and propose an efficient and partial policy hiding ciphertext-policy ABE scheme suitable for VFC. Experiments have proved that our scheme is computationally more efficient than previous policy hiding ABE schemes.

Keywords Vehicular fog computing · CP-ABE · Partial policy hiding · Prime order bilinear group

1 Introduction

With the popularity of 5G communications, research on the application of the Internet of Things (IoT) has become increasingly popular. More and more IoT applications appear in people's daily lives, such as smart medical care, smart cities, and vehicular ad hoc network (VANET). Lee et al. (2016) proposed a key agreement technology to the vehicular ad hoc network (VANET) communication channels. Vehicular fog computing (VFC) (Huang et al. 2017) is one of the research focuses, which combines traditional VANET with fog computing and further utilizes the computing power of these fog nodes of RSU to meet the real-time and high-efficiency requirements in the application. Due to the particularity of VFC, the security requirements that need to achieve are also different from other applications. In VFC, in addition to confidentiality and access control that need to be considered in the general environment, the algorithms in the designed security protocol must be sufficiently efficient for the limitation of computing power. Therefore, it is necessary to construct a secure and efficient encryption scheme with an access control function to meet the needs of VFC.

The traditional public-key encryption scheme can only meet the confidentiality requirements because the ciphertext can only be decrypted via a private key corresponding to the public key used for encryption. This is a one-to-one encryption scheme with no access control. Attribute-based encryption (ABE) adds access control functions to traditional public-key encryption, which not only satisfies confidentiality but also realizes access control. ABE includes ciphertext-policy ABE (CP-ABE) and key-policy ABE (KP-ABE) (Tian et al. 2020). In CP-ABE schemes, the ciphertext is corresponding to the access policy, and the secret key is corresponding to a set of attributes, while in KP-ABE schemes, the structure is just the opposite. Only when a certain subset of the attribute set meets the access policy decryption can succeed. CP-ABE is more suitable for VFC because users can dynamically specify the access control structure required for decryption when encrypting, which is more flexible. Feng et al. (2020), Alrawais et al. (2017), Jiang et al. (2018) already have applied ABE to edge computing and VANET.

Although traditional CP-ABE can realize confidentiality and flexible access control functions, there are still some

✉ Yongjian Liao
liaoyj@uestc.edu.cn

[1] University of Electronic Science and Technology of China, Chengdu, China

in-depth issues that need to be considered. The openness of the access policy is an implicit problem of the traditional CP-ABE. In the traditional CP-ABE, the plaintext of the access policy with sensitive information will be sent to another user together with the ciphertext of the message, which may threaten the user's privacy. For example, suppose Alice uses CP-ABE to encrypt a message, and the specified access policy is "(driving age: more than 4 years **AND** location: Chengdu) **OR** (gender: male **AND** vehicle type: truck)". Bob with the attribute set "gender: male, vehicle type: truck, driving age: 3 years" can decrypt and obtain the plaintext information, while with the attribute set of "driving age: 5 years, gender: female, vehicle type: car ", Lina cannot decrypt. However, we found that although Lina cannot decrypt, she can obtain sensitive information such as the decryptor's address and vehicle type.

According to the problems mentioned above, the concept of "policy hiding" was put forward. "policy hiding" is divided into two types: "full policy hiding" and "partial policy hiding". The CP-ABE scheme of "full policy hiding" will hide all the attribute information in the policy. Only through decryption we can know whether the individual attribute set meets the access policy, but the decryptor cannot obtain any information in the policy. Katz et al. (2008) proposed a CP-ABE based on the inner-product predicate encryption (IPE) structure, but this scheme can only use a threshold access structure, which is far less flexible than the commonly used linear secret sharing scheme (LSSS). CP-ABE scheme of "partial policy hiding" will hide part of the attribute information. Although some of the attribute information will be exposed, it does not affect the overall security. Nishide et al. (2008) proposed a CP-ABE scheme with "partial policy hiding", but their scheme only supports AND-GATE access structure, and it is only selective security. Zhang et al. (2018) proposed another CP-ABE scheme with "partial policy hiding". They divide the attributes into attribute names and attribute values, and the access policies will expose the attribute names, while the corresponding attribute values were hidden in the access policy. As in the above example, the information in the access policy that Lina can obtain is "(driving age: - **AND** location: -) **OR** (gender: - **AND** vehicle type: -)". The adversary only obtains the name of the attribute that is not sensitive, and the sensitive and specific attribute values are protected. This scheme achieves complete security and "partial policy hiding" and adds a decryption test before decryption, which further improves the efficiency of decryption. However, this scheme is proposed based on the composite order group whose overload of computation is heavy. Guillevic (2013) had compared the computation on the composite order group and prime order group and recommended to use the prime order group. Therefore, the scheme of Zhang et al. (2018) is not suitable for VFC scenarios. Our target is to construct a prime-order-group-based, adaptive secure, and large universe CP-ABE scheme for VFC.

1.1 Our contribution

According to the problems described above, we construct an adaptive secure, prime-order-group-based, and large universe CP-ABE. We used the technology of Freeman (2010) and proposed a CP-ABE scheme suitable for VFC. The detailed advantages of this scheme are:

1. The use of prime-order group bilinear pair group greatly reduces computation load under the same security and satisfies the application scenarios of VFC, which are limited in computation. There is a decryption test phase before the decryption phase, which further improves the efficiency of decryption;
2. Separate the attribute value from the attribute name. Only the insensitive attribute name is included in the access control structure, and the attribute value is hidden in the ciphertext. This method hides the users sensitive information and protects user privacy;
3. Our scheme is large universe, which means the size of the attribute universe can be exponentially large and the size of public parameters is constant. In most of the previous schemes, the size of public parameters grows linearly with the size of the universe;
4. Our scheme can be proved adaptive security under the standard model. Compared with selective security, an adaptive security scheme is more usable and more secure;

1.2 Related work

Recently, many works about VFC have been proposed. Xiao and Zhu (2017) presented a visionary concept called VFC. They proposed the VFC architecture and some related requirements. Ning et al. (2019) presented a VFC-enabled traffic management scheme for smart cities. They constructed a three-layer VFC architecture to dynamically cooperate with each other for network load balancing. They also emphasized the security issues faced in VFC. Hou et al. (2016) presented a new paradigm referred to as VFC. They added the vehicle nodes to the fog node, making full use of the computing power of the vehicle. Huang et al. (2017) put forward the common architecture of VFC and its security requirements.

Sahai and Waters (2005) proposed the concept of ABE. Then, Goyal et al. (2006) constructed the first KP-ABE scheme in 2006, and Bethencourt et al. (2007) constructed the first CP-ABE scheme in 2007. Lee et al. (2013) made a comprehensive survey of CP-ABE and KP-ABE. Liao et al. (2020) and Chen and Liao (2019) proposed two outsourced attribute-based encryption schemes. Nishide et al. (2008) firstly proposed a "partial policy hiding" ABE scheme. They

used inner-product predicate encryption to hide the attribute in policy, and their scheme only supports an AND-GATE access structure. Lai et al. (2011) improved the work of Nishide et al. (2008) and proposed a "full policy hiding" ABE with the same access structure. However, the size of the ciphertext grows with the number of attributes. Yang et al. (2016) constructed an adaptive secure CP-ABE scheme with an AND-GATE access structure, but only supported small universe and used the composite order group. The scheme of Zhang et al. (2013) was built on prime order group, but can only be proved selective security. Schemes of Zhao et al. (2019) and Zhang et al. (2019) supported policy tree, which is more flexible than AND-GATE, but both of them supported small universe. Lai et al. (2012) proposed an adaptive security CP-ABE that supports large universe with a decryption test. Their scheme used the composite order group, which is not efficient. Zhang et al. (2018) improved the scheme of Lai et al. (2012) and further improved efficiency. However, their scheme also used the composite order group, which is very inefficient compared to the prime order group.

1.3 Organization

The organization of our paper is as follows: Section 2 introduces some definitions used in our system. Section 3 introduces our system and CP-ABE scheme. In Sect. 4, we prove the security of our scheme. Section 5 presents the efficiency analysis. In Sect. 6, we conclude our work.

2 Preliminaries

In this section, we will introduce the bilinear group generator used in our scheme. Besides, we will introduce the assumption used in our proof.

2.1 Bilinear groups

Bilinear Groups Generator Bilinear group generator ($\mathcal{G}$) takes a security parameter λ as input and outputs a set of groups and a pairing $G, G^{'}, H, H^{'}, G_t, \hat{e} : G \times H \rightarrow G_t$, where $G^{'} \subset G$ and $H^{'} \subset H$. The pairing must satisfy the following properties:

- Bilinear: for all $g_1, g_2 \in G$ and $h_1, h_2 \in H$, we have $\hat{e}(g_1 g_2, h_1 h_2) = \hat{e}(g_1, h_1)\hat{e}(g_1, h_2)\hat{e}(g_2, h_1)\hat{e}(g_2, h_2)$;
- Nondegenerate: for any $g \in G$ (or $h \in H$), and for all $h \in H$ (or $g \in G$), we have $\hat{e}(g, h) = 1$, then $g = 1$ (or $h = 1$);
- Computable: for any $g \in G$ and $h \in H$, we can calculate $\hat{e}$ in polynomial time.

Cancelling pairing bilinear group generator According to work of Freeman (2010), we construct our bilinear group, whose components are of prime order. Choose a symmetric pairing, we can obtain a $4 - Cancell\ ing$ bilinear group generator $\mathcal{G}_L$ (5, 2) as follows:

1. Let $(p, \mathbb{G}, \mathbb{G}_t, e) \leftarrow \mathcal{P}(\lambda)$. $\mathcal{P}$ denotes a prime order bilinear group generator. Then set $G = H = \mathbb{G}^5$ and $G_t = \mathbb{G}_t$;
2. Choose $\mathbf{x}_1, \ldots, \mathbf{x}_5 \leftarrow \mathbb{F}_p^5$, where the vectors $\{\mathbf{x}_i\}$ are linearly independent of each other. For $2 < i \leq 5$ and $1 \leq j \leq 5$, we require that if $i \neq j$ then $\mathbf{x}_i \cdot \mathbf{x}_j = 0$, and if $i = j$, then $\mathbf{x}_i \cdot \mathbf{x}_j \neq 0$;
3. $g \leftarrow \mathbb{G}$ is a random generator. We set $\boldsymbol{\theta}_i = g^{\mathbf{x}_i} \in G$;
4. We define:

$$\boldsymbol{\theta}_i \boldsymbol{\theta}_j = \left(g^{x_{i,1}} g^{x_{j,1}}, g^{x_{i,2}} g^{x_{j,2}}, \ldots, g^{x_{i,5}} g^{x_{j,5}}\right).$$

   The symbol $\langle X \rangle$ represents the group generated by a finite set X. Let $G_1 = \langle \boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \rangle$, and $G_i = \langle \boldsymbol{\theta}_{i+1} \rangle$ for $2 \leq i \leq 4$;
5. Define the pairing e as $e((g_1, \ldots, g_5), (h_1, \ldots, h_5)) = \sum_{i=1}^{5} \hat{e}(g_i, h_i)$, where $(g_1, \ldots, g_5)$ and $(h_1, \ldots, h_5)$ are elements of G_i and G_j respectively;
6. Output $(G, G_1, \ldots, G_4, G_t, e)$;

Choose $\mathbf{g}_i \in_R G_i, \mathbf{h}_j \in_R G_j$ .We find that $e(\mathbf{g}_i, \mathbf{h}_j) = 1$ if $i \neq j$ and $e(\mathbf{g}_i, \mathbf{h}_j) \neq 1$ if $i = j$.

2.2 Assumptions

Subgroup decision problem $\mathcal{G}$ is the bilinear group generator introduced above. We define the distribution as follows:

$$\mathbb{G} = \left(G, G^{'}, H, H^{'}, G_t, e\right) \leftarrow \mathcal{G}, T_0 \leftarrow G, T_1 \leftarrow G^{'}.$$

If there exists an algorithm $\mathcal{A}$ that can solve the *subgroup decision problem on the left*. $SDP_L\text{-}Adv[\mathcal{A}, \mathcal{G}]$ denotes the advantage to solve the *subgroup decision problem on the left*:

$$= |Pr[\mathcal{A}(\mathbb{G}, T_0) = 1] - Pr[\mathcal{A}(\mathbb{G}, T_1) = 1]|.$$

If $SDP_L\text{-}Adv[\mathcal{A}, \mathcal{G}]$ is a negligible function of λ, then $\mathcal{G}$ satisfies the *subgroup decision assumption on the left*. Analogously, if we define $T_0 \leftarrow H$ and $T_1 \leftarrow H^{'}$, we can define the *subgroup decision assumption on the right*. If $\mathcal{G}$ satisfies both assumptions, we call $\mathcal{G}$ satisfies the *subgroup decision assumption*.

k-Linear assumption If groups G, G_1, H, H_1, G_t generated by $\mathcal{P}$ all have prime order $p > 2^{\lambda}$, we call $\mathcal{P}$ is a prime-order bilinear group generator. For all groups generated by $\mathcal{P}$ have the same prime order, we have $G = G_1$ and $H = H_1$. We use $\mathbb{G}_1 = G$, $\mathbb{G}_2 = H$, and $\mathbb{G}_t = G_t$ to denote the three distinct groups. Let $\hat{\mathbb{G}}$ denote the output

Table 1 Notation table

Notations	Descriptions
$a \in_R A$	The element a is randomly chosen from the set A
$\mathcal{G}_L(5, 2)$	The cancelling pairing bilinear group generator described in Sect. 2.1.
G, G_t	Two cyclic multiplicative groups
G_i	A subgroup of G with prime order p
SK_θ	A secret key associated with an attribute set θ
$\theta = (I_S, S)$	I_S denotes the attribute name index
	S is the attribute value set
$\mathrm{CT}_{\mathbb{A}}$	A CP-ABE ciphertext associated with an access policy $\mathbb{A}$
$\mathbb{A} = (A, \rho, T)$	A is the access policy matrix
	ρ maps the index of row in A to an attribute name
	T is an attribute value set
$\mathcal{I}$	A minimum authorized set of access policy (A, ρ)
$\theta_{A,\rho}$	The set of $\mathcal{I}$

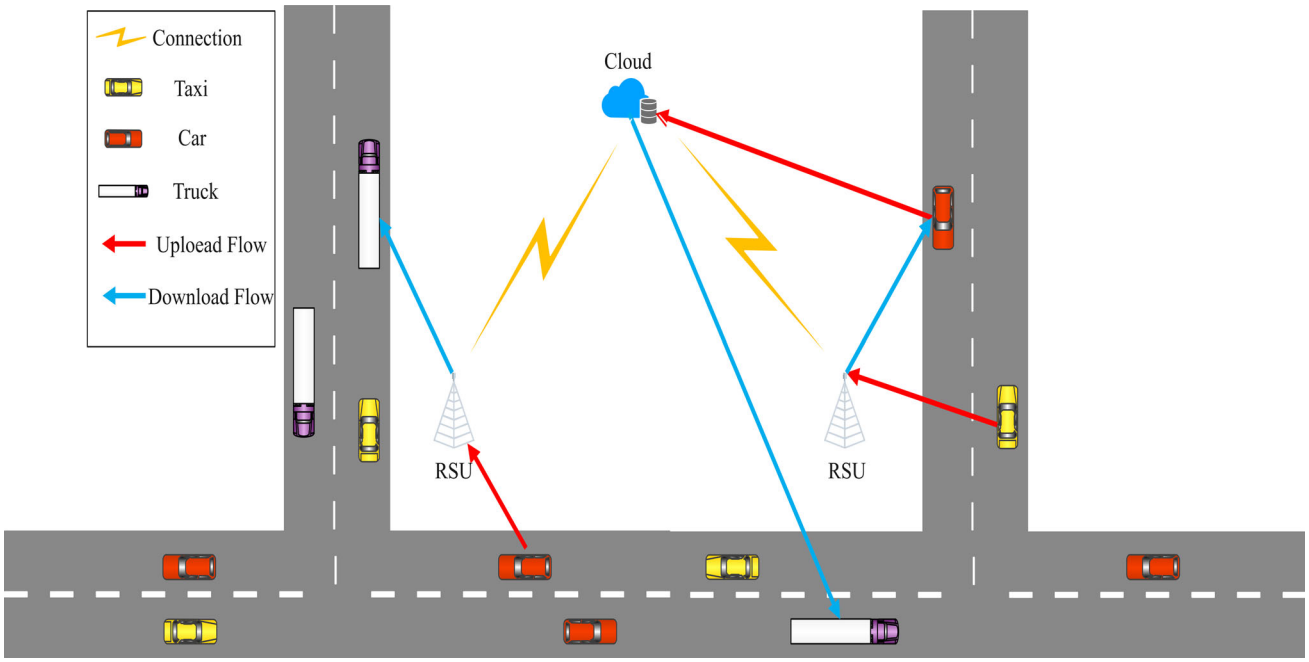

Fig. 1 System model

$(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, e)$ of $\mathcal{P}(\lambda)$, $k \geq 1$ be an integer. We define the advantage of an algorithm $\mathcal{A}$ in solving the k-Linear problem in $\mathbb{G}_1$ as $k\text{-}Lin_{\mathbb{G}_1}\text{-}Adv[\mathcal{A}, \mathcal{P}]$:

$$\left| Pr\left[\mathcal{A}\left(\hat{\mathbb{G}}, g_1, \ldots, g_k, g_1^{r_1}, \ldots, g_k^{r_k}, h, h^{r_1+\cdots+r_k}\right) = 1\right] - Pr\left[\mathcal{A}\left(\hat{\mathbb{G}}, g_1, \ldots, g_k, g_1^{r_1}, \ldots, g_k^{r_k}, h, h^{s}\right) = 1\right] \right|.$$

Similarly for $k\text{-}Lin_{\mathbb{G}_2}\text{-}Adv[\mathcal{A}, \mathcal{P}]$. We say that $\mathcal{G}$ satisfies the k-Linear assumption in $\mathbb{G}_1$ if $k\text{-}Lin_{\mathbb{G}_1}\text{-}Adv[\mathcal{A}, \mathcal{P}](\lambda)$ is a negligible function of λ for any polynomial-time algorithm $\mathcal{A}$ (Similarly for $\mathbb{G}_2$).

Lemma 1 *$\mathcal{P}$ satisfies the k-Linear assumption in $\mathbb{G}$; then, $\mathcal{G}_L(5, 2)$ satisfies the subgroup decision assumption. The proof of this lemma can be found in Theorem 2.5 of* Freeman (2010).

3 System

In this section, firstly, we introduce our system model. Then, we put forward the security and performance requirements of our system. Finally, we give the detail of our scheme. Table 1 lists the notation table for the symbols in our system.

3.1 System model

In this subsection, we will introduce the system architecture of our scheme. We propose a partial policy hiding CP-ABE based on the prime order group and use it in VFC to construct a secure VFC model that can meet real-time requirements. There are four entities in our system.

1. *Key Generation Center (KGC)* KGC is a trusted center that is responsible for system initialization and key generation. KGC generates public parameters and the master private key according to the security parameter and then distributes public parameters. KGC can also generate the corresponding private key according to the user's attribute set and send it to the user. Cloud center (CC), road side unit (RSU), and vehicle object (VO) can register and authenticate at KGC to obtain their own private key;
2. *CC* CC, with large storage capacity and computing power, can process and store a large amount of data uploaded by VO and RSU. CC can register at KGC to obtain its own private key. If VO adds CC's attributes to the access policy, CC can decrypt and process these data;
3. *RSU* RSU has limited storage and computing power that is weaker than CC. RSU is mainly responsible for processing the data uploaded by vehicle object with high real-time performance requirements. RSU can register at KGC to obtain its own private key, and vehicle object can specify the attributes of RSU in the policy so that RSU can decrypt and process data. RSU can also process the vehicle object data and forward the encrypted one to CC for processing and storage;
4. *VO* VO generates data and designs the policy to encrypt the data and upload it to RSU or CC. VO can register with KGC to obtain its own private key. VO can download data from the cloud or RSU and decrypt it with its own private key. The computing power of VO is very poor;

Figure 1 shows our system model. VO includes trucks, cars, and taxis. Cloud is CC and RSU is the roadside unit described above. In the system, KGC runs **Setup** to generate public key and master secret key. Then, KGC can run **KeyGen** to generate secret keys according to the user's attribute sets. VO specifies the policy and runs **Encryption** to encrypt data and then publishes the encrypted data to RSU or CC. VO, CC and RSU can use the secret key received from KGC to run **Decryption** to get the decrypted data. Figure 2 shows the detailed data flow in our system.

3.2 Security and performance requirement

In this subsection, we will propose the security and performance requirements that our system meets. In our system,

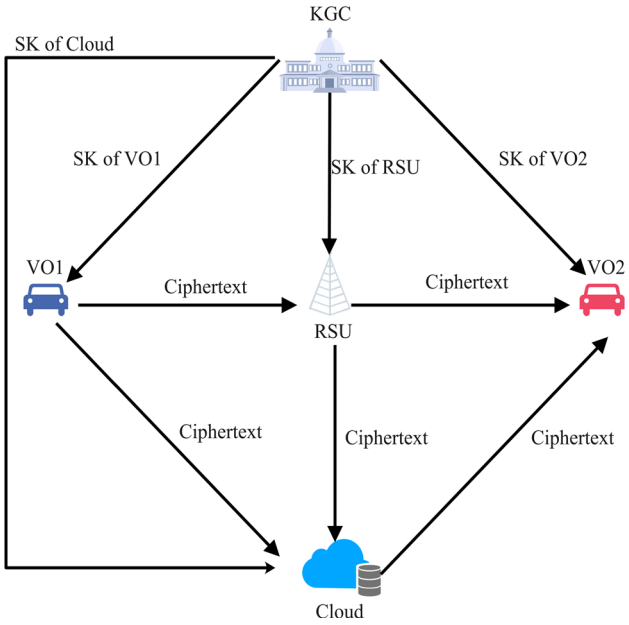

Fig. 2 Data flow

both CC and RSU are curious but honest, that is, they both will transmit and process data honestly but hope to get the secret information of VO. We list the security requirement as follows:

1. *Privacy of Plaintext* The ciphertext will perfectly hide the information about the plaintext. Adversary cannot get any information except the length of the plaintext without decryption;
2. *Collusion Resistance* Any user, whose attribute sets do not satisfy the policy, respectively, cannot decrypt the ciphertext, even a collection of their attributes satisfies the policy. For example, Alice has the secret key for attribute set "('Number Plate': odd), ('Model': truck)", while Bob has the secret key for attribute set "('Number Plate': even), ('Model': car)". They are not satisfied with the policy "('Number Plate': even) AND ('Model': truck)", respectively, so they cannot decrypt the ciphertext although the collection of their attribute set satisfies the policy;
3. *Partially Policy Hiding* The policy sent with ciphertext only exposes the information about attribute name, but doesn't expose the specific attribute contents which satisfy the policy. In the actual environment, the specific value of the attributes often contains a lot of sensitive information of the user, and the attribute names are not sensitive;

Then, we introduce the performance requirements as follows:

1. *Large Universe* Large universe means that the size of the public parameter has nothing to do with the size of the attribute universe. In the small universe scheme, the size of the public parameter increases linearly with the size of the attribute domain, which means that we must fix a very large public parameter at system initialization. In our system, there are a large number of attributes of CC, RSU, and VO, so our system meets the large universe;
2. *Efficient Decryption* In our system, VO has poor computing power. Although the computing power of RSU is stronger than that of VO, it is also insufficient. Therefore, we should minimize the amount of calculation in the decryption step to reduce the computational amount of VO and RSU and reduce the calculation time;

3.3 Detail of CP-ABE scheme

KGC uses $\mathcal{G}_L$ (5, 2) to generate $(G, G_1, G_2, G_3, G_4, G_t, e)$. The order of G is $N = p^5$, and G_1, G_2, G_3, G_4 are subgroups of G whose element is all of prime order p. The detail of our CP-ABE scheme is as follows:

1. *Setup*$\left(1^\lambda\right)$ KGC inputs the security parameter and then gets the public parameters PK and the master secret key MSK. The attribute set is $U = \mathbb{Z}_p$. KGC picks random elements $a, b \in_R \mathbb{Z}_p$, $\mathbf{g}_1, \mathbf{h}_1 \in_R G_1$ and $\mathbf{Z}_4, \mathbf{g}_4 \in_R G_4$, sets $Y = e\left(\mathbf{g}_1, \mathbf{g}_1\right)^a$ and $\mathbf{Z} = \mathbf{h}_1\mathbf{Z}_4$, chooses $\mathbf{g}_3 \in G_3$ uniformly. Finally, the **Setup** algorithm outputs the PK and MSK:

$$\mathrm{PK} = \left(p, \mathbf{g}_1, \mathbf{g}_1^b, Y, \mathbf{Z}, \mathbf{g}_4\right)$$
$$\mathrm{MSK} = \left(a, \mathbf{h}_1, \mathbf{g}_3\right).$$

2. *KeyGen*(PK, MSK, θ) KGC receives the user's (VO or RSU or CC) attribute set θ and returns the secret key SK_θ associated with the attribute set to the corresponding user. The attribute set of the user is $\theta = (I_S, S)$, where $I_S \in \mathbb{Z}_p$ is the attribute name index, and $S = \{s_i\}_{i \in I_S}$ is the set of attribute values. KGC picks random number $r \in_R \mathbb{Z}_p$, then randomly chooses $\mathbf{R}_3, \mathbf{R}_3', \mathbf{R}_{3,i} \in_R G_3$ from $\mathbf{g}_3$ where $i \in I_S$. Finally, algorithm can output user's secret key:

$$\mathrm{SK}_\theta = \left(\theta, K, K', \{K_i\}\right),$$

where

$$K = \mathbf{R}_3\left(\mathbf{g}_1^a\mathbf{g}_1^{br}\right), K' = \mathbf{R}_3'\mathbf{g}_1^r, K_i = \mathbf{R}_{3,i}\left(\mathbf{g}_1^{s_i}\mathbf{h}_1\right)^r.$$

3. *Encryption*(PK, M, $\mathbb{A}$) VO sets the access policy $\mathbb{A}$ and runs the **Encryption** algorithm to generate the ciphertext $\mathrm{CT}_\mathbb{A}$, then sends the ciphertext with an access policy to RSU or CC. In the input, the $M \in G_T$ denotes the plaintext. $\mathbb{A} = (A, \rho, T)$ is an access policy, where A is a matrix with ℓ rows and n columns. ρ is a map from each row of A_j to the attribute name, and $T = \left(t_{\rho(1)}, t_{\rho(2)}, \ldots, t_{\rho(\ell)}\right) \in \mathbb{Z}_p^\ell$ is the set of attribute values. VO randomly chooses two vectors $\mathbf{v}_1, \mathbf{v}_2 \in_R Z_p^N$, $\mathbf{v}_1 = \left(s, v_{1,2}, \ldots, v_{1,n}\right)$ and $\mathbf{v}_2 = (s', v_{2,2}, \ldots, v_{2,n})$, then randomly choose $\mathbf{X}_2, \mathbf{X}_{2,j}, \mathbf{X}_{1,j}, \mathbf{X}_{1,j}' \in_R G_4$ based on $\mathbf{g}_4$ and $r_j \in_R \mathbb{Z}_p$, where $1 \le j \le \ell$. Finally, **Encryption** algorithm outputs the ciphertext:

$$\mathrm{CT}_\mathbb{A} = \left((A, \rho), C_1, C_1', D_{1,j}, D_{1,j}', C_2, C_2', D_{2,j}\right),$$

where

$$C_1 = M \cdot Y^s,\ D_{1,j}' = \mathbf{g}_1^{r_j}\mathbf{X}_{1,j}',\ C_1' = \mathbf{g}_1^s,$$
$$D_{1,j} = \mathbf{g}_1^{bA_j \cdot \mathbf{v}_1}\left(\mathbf{g}_1^{t_{\rho(j)}}\mathbf{Z}\right)^{-r_j}\mathbf{X}_{1,j}$$
$$C_2 = Y^{s'},$$
$$C_2' = \mathbf{g}_1^{s'}\mathbf{X}_2,\ D_{2,j} = \mathbf{g}_1^{bA_j \cdot \mathbf{v}_2}\left(\mathbf{g}_1^{t_{\rho(j)}}\mathbf{Z}\right)^{-s'}\mathbf{X}_{2,j}.$$

4. *Decryption*(PK, $\mathrm{CT}_\mathbb{A}$, SK_θ) User (VO or RSU or CC) firstly checks whether their keys can decrypt the ciphertext. If they pass the test phase, then they can enter the final decryption phase to recover the plaintext.

(a) *Test* Firstly, users calculate $\theta_{A,\rho}$ from (A, ρ). Then, it checks if there exists a subset $\mathcal{I} \in \theta_{A,\rho}$ that satisfies $\{\rho(i) | i \in \mathcal{I}\} \subseteq I_S$. If no such subset, the algorithm outputs $\perp$ denoted the user's attribute names do not satisfy the access policy. If there exists such a subgroup, users can then calculate a set of constants $\{\omega_i\}$ which satisfies $\sum_{i \in \mathcal{I}} \omega_i A_i = (1, 0, \ldots, 0)$. Then, users can check:

$$C_2^{-1} = e\left(\prod_{i \in \mathcal{I}} D_{2,i}^{\omega_i}, K'\right) e\left(C_2', K^{-1}\prod_{i \in \mathcal{I}} K_{\rho(i)}^{\omega_i}\right).$$

If this equation holds, users can decrypt to get plaintext, else output $\perp$.

(b) *Final Decryption* Firstly, users calculate:

$$E = \frac{e\left(C_1', K\right)}{\prod_{i \in \mathcal{I}}\left(e\left(D_{1,i}, K'\right), e\left(D_{1,i}', K_{\rho(i)}\right)\right)^{\omega_i}}$$

Then users can recover the plaintext: $M = C_1/E$.

3.4 Correctness of scheme

The correctness of CP-ABE means that the user can decrypt to recover the plaintext only when his attribute set satisfies the access policy. For the **Test** phase in **Decryption**, if the user's attribute set satisfies the access policy, then we have:

$$C_2^{-1} = e\left(\prod_{i\in\mathcal{I}} D_{2,i}^{\omega_i}, K^{'}\right) e\left(C_2^{'}, K^{-1}\prod_{i\in\mathcal{I}} K_{\rho(i)}^{\omega_i}\right) = \frac{\prod_{i\in\mathcal{I}}\left(e\left(D_{2,i}, K^{'}\right) e\left(C_2^{'}, K_{\rho(i)}\right)\right)^{\omega_i}}{e\left(C_2^{'}, K\right)}$$

If and only if $t_{\rho(i)} = s_{\rho(i)}$, for $i \in \mathcal{I}$, we have:

$$\begin{aligned}
&\prod_{i\in\mathcal{I}}\left(e\left(D_{2,i}, K^{'}\right) e\left(C_2^{'}, K_{\rho(i)}\right)\right)^{\omega_i}\\
=&\prod_{i\in\mathcal{I}} e\left(\mathbf{g}_1^{bA_i\cdot\mathbf{v}_2}\mathbf{X}_{2,i}\left(\mathbf{g}_1\mathbf{Z}\right)^{-s^{'}}, \mathbf{R}_3^{'}\mathbf{g}_1^{r}\right)^{\omega_i}\cdot\\
&\prod_{i\in\mathcal{I}} e\left(\mathbf{g}_1^{s^{'}}\mathbf{X}_2, \left(\mathbf{g}_1^{s_{\rho(i)}}\mathbf{h}_1\right)^{r}\mathbf{R}_{3,\rho(i)}\right)^{\omega_i}\\
=&\prod_{i\in\mathcal{I}} e\left(\mathbf{g}_1, \mathbf{g}_1\right)^{rb\omega_i A_i\cdot\mathbf{v}_2}\\
=&e\left(\mathbf{g}_1, \mathbf{g}_1\right)^{\sum_{i\in\mathcal{I}}\omega_i A_i rb\cdot\mathbf{v}_2} = e\left(\mathbf{g}_1, \mathbf{g}_1\right)^{s^{'}br}
\end{aligned}$$

and

$$e\left(C_2^{'}, K\right) = e\left(\mathbf{g}_1^{s^{'}}\mathbf{X}_2, \mathbf{R}_3\mathbf{g}_1^{a}\mathbf{g}_1^{br}\right) = e\left(\mathbf{g}_1, \mathbf{g}_1\right)^{as^{'}+brs^{'}}.$$

Finally we have:

$$\frac{e\left(\mathbf{g}_1, \mathbf{g}_1\right)^{as^{'}+brs^{'}}}{e\left(\mathbf{g}_1, \mathbf{g}_1\right)^{s^{'}br}} = e\left(\mathbf{g}_1, \mathbf{g}_1\right)^{as^{'}} = C_2.$$

Similarly, for the **Final Decryption** phase in **Decryption**, we have:

$$E = \frac{e\left(C_1^{'}, K\right)}{\prod_{i\in\mathcal{I}}\left(e\left(D_{1,i}, K^{'}\right), e\left(D_{1,i}^{'}, K_{\rho(i)}\right)\right)^{\omega_i}}$$

Then, we can calculate:

$$\begin{aligned}
&\prod_{i\in\mathcal{I}}\left(e\left(D_{1,i}, K^{'}\right), e\left(D_{1,i}^{'}, K_{\rho(i)}\right)\right)^{\omega_i}\\
=&\prod_{i\in\mathcal{I}} e\left(\mathbf{g}_1^{bA_i\cdot\mathbf{v}_1}\left(\mathbf{g}_1^{t_{\rho(i)}}\mathbf{Z}\right)^{-r_i}\mathbf{X}_{1,i}, \mathbf{R}_3^{'}\mathbf{g}^{r}\right)^{\omega_i}\cdot\\
&\prod_{i\in\mathcal{I}} e\left(\mathbf{g}_1^{r_i}\mathbf{X}_{1,i}^{'}, \mathbf{R}_{3,\rho(i)}\left(\mathbf{g}_1^{s_{\rho(i)}}\mathbf{h}_1\right)^{r}\right)^{\omega_i}\\
=&\prod_{i\in\mathcal{I}} e\left(\mathbf{g}_1, \mathbf{g}_1\right)^{rb\omega_i A_i\cdot\mathbf{v}_1}\\
=&e\left(\mathbf{g}_1, \mathbf{g}_1\right)^{rb\mathbf{v}_1\cdot\sum_{i\in\mathcal{I}}\omega_i A_i} = e\left(\mathbf{g}_1, \mathbf{g}\right)^{brs}
\end{aligned}$$

and

$$e\left(C_1^{'}, K\right) = e\left(\mathbf{g}_1^{s}, \mathbf{R}_3\left(\mathbf{g}_1^{a}\mathbf{g}_1^{br}\right)\right) = e\left(\mathbf{g}_1, \mathbf{g}_1\right)^{as+brs}.$$

Finally, we can calculate $E = \frac{e(\mathbf{g}_1,\mathbf{g}_1)^{as+brs}}{e(\mathbf{g}_1,\mathbf{g})^{brs}} = e\left(\mathbf{g}_1, \mathbf{g}_1\right)^{as} = Y^s$ and then recover the plaintext $C_1/E = \frac{M\cdot Y^s}{Y^s} = M$.

4 Security proof

In this section, we firstly give our security model. Then, we introduce our assumption and the proving process. Finally, we analyze the security for our VFC system.

4.1 Security model for CP-ABE

In this section, we will define the adaptive security for our scheme. The game between an adversary $\mathcal{A}$ and a challenger $\mathcal{B}$ is

1. *Setup* $\mathcal{B}$ executes $\mathbf{Setup}\left(1^{\lambda}\right)$ to get the public key PK and master secret key MSK. Then, $\mathcal{B}$ sends PK to $\mathcal{A}$;
2. *Phase1* $\mathcal{A}$ submits some attribute sets $\theta = (I_S, S)$. $\mathcal{B}$ runs $\mathbf{KeyGen}(\mathrm{PK}, \mathrm{MK}, \theta)$ to generate secret keys SK_θ and transmits to $\mathcal{A}$;
3. *Challenge* $\mathcal{A}$ chooses two messages M_0 and M_1. Then, $\mathcal{A}$ chooses two access policies $\mathbb{A}_0 = (A, \rho, T_0)$ and $\mathbb{A}_1 = (A, \rho, T_1)$ and sends the access policies with the messages to $\mathcal{B}$. Both access policies should not be satisfied by the attribute sets queried in **Phase1**. $\mathcal{B}$ randomly chooses $b \in \{0, 1\}$. Then, $\mathcal{B}$ runs $\mathbf{Encryption}(\mathrm{PK}, M_b, \mathbb{A}_b)$ to genenrate the ciphertext $\mathrm{CT}_{\mathbb{A}_b}$. Finally, $\mathcal{B}$ sends $\mathrm{CT}_{\mathbb{A}_b}$ to $\mathcal{A}$;
4. *Phase2* The same as **Phase1**, except the queried attribute sets should not satisfy the $\mathbb{A}_0$ and $\mathbb{A}_1$ in **Challenge**;
5. *Guess* $\mathcal{A}$ guesses $b^{'} \in \{0, 1\}$. If $b^{'} = b$, $\mathcal{A}$ wins the game.

The advantage of $\mathcal{A}$ winning the game is defined as $\mathrm{Adv}_{\mathcal{A}} = \left|\Pr\left(b^{'} = b\right) - 1/2\right|$.

4.2 Assumptions

According to lemma 1, $\mathcal{G}_L$ (5, 2) satisfies the subgroup decision assumption. Then, we have the following assumptions.

Assumption 1 For the group generator $\mathcal{G}_L(5,2)$, define the following distribution:

$$(p, G, G_t, e) \xleftarrow{R} \mathcal{G}_L(5,2), \mathbf{g}_1 \xleftarrow{R} G_1, \mathbf{P}_3 \xleftarrow{R} G_3, \mathbf{P}_4 \xleftarrow{R} G_4.$$

Then we choose:

$$D = (p, G, G_t, e, \mathbf{g}_1, \mathbf{P}_3, \mathbf{P}_4), \mathbf{X}_1 \xleftarrow{R} \langle G_1, G_2 \rangle, \mathbf{X}_2 \xleftarrow{R} G_1.$$

The advantage for an algorithm $\mathcal{A}$ to distinguish $\mathbf{X}_1$ and $\mathbf{X}_2$ is defined as:

$$\mathrm{Adv1}_{\mathcal{G}_L(5,2),\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(D, \mathbf{X}_1) = 1] - Pr[\mathcal{A}(D, \mathbf{X}_2) = 1]|.$$

If $\mathrm{Adv1}_{\mathcal{G}_L(5,2),\mathcal{A}}(\lambda)$ is negligible, the group generator $\mathcal{G}_L(5,2)$ satisfies Assumption 1.

Assumption 2 For the group generator $\mathcal{G}_L(5,2)$, define the following distribution:

$$(p, G, G_t, e) \xleftarrow{R} \mathcal{G}_L(5,2), \mathbf{g}_1, \mathbf{P}_1 \xleftarrow{R} G_1, \mathbf{P}_2, \mathbf{Q}_2 \xleftarrow{R} G_2, \mathbf{P}_3, \mathbf{Q}_3 \xleftarrow{R} G_3, \mathbf{P}_4 \xleftarrow{R} G_4.$$

Then we choose:

$$D = (p, G, G_t, e, \mathbf{g}_1, \mathbf{P}_1\mathbf{P}_2, \mathbf{Q}_2\mathbf{Q}_3, \mathbf{P}_3, \mathbf{P}_4), \mathbf{X}_1 \xleftarrow{R} \langle G_1, G_2, G_3 \rangle, \mathbf{X}_2 \xleftarrow{R} \langle G_1, G_3 \rangle.$$

The advantage for an algorithm $\mathcal{A}$ to distinguish $\mathbf{X}_1$ and $\mathbf{X}_2$ is defined:

$$\mathrm{Adv2}_{\mathcal{G}_L(5,2),\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(D, \mathbf{X}_1) = 1] - \Pr[\mathcal{A}(D, \mathbf{X}_2) = 1]|.$$

If $\mathrm{Adv2}_{\mathcal{G}_L(5,2),\mathcal{A}}(\lambda)$ is negligible, the group generator $\mathcal{G}_L(5,2)$ satisfies Assumption 2.

Assumption 3 For the group generator $\mathcal{G}_L(5,2)$, define the following distribution:

$$(p, G, G_t, e) \xleftarrow{R} \mathcal{G}_L(5,2), a, b \xleftarrow{R} \mathbb{Z}_p$$
$$\mathbf{g}_1 \xleftarrow{R} G_1, \mathbf{g}_2, \mathbf{P}_2, \mathbf{Q}_2 \xleftarrow{R} G_2, \mathbf{P}_3 \xleftarrow{R} G_3, \mathbf{P}_4 \xleftarrow{R} G_4.$$

Then, we choose:

$$D = \left(p, G, G_t, e, \mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_1^a\mathbf{P}_2, \mathbf{g}_1^b\mathbf{Q}_2, \mathbf{P}_3, \mathbf{P}_4\right),$$
$$X_1 = e(\mathbf{g}_1, \mathbf{g}_1)^{ab}, X_2 \xleftarrow{R} G_t.$$

The advantage for an algorithm $\mathcal{A}$ to distinguish X_1 and X_2 is defined:

$$\mathrm{Adv3}_{\mathcal{G}_L(5,2),\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(D, X_1) = 1] - \Pr[\mathcal{A}(D, X_2) = 1]|.$$

If $\mathrm{Adv3}_{\mathcal{G}_L(5,2),\mathcal{A}}(\lambda)$ is negligible, the group generator $\mathcal{G}_L(5,2)$ satisfies Assumption 3.

Assumption 4 For the group generator $\mathcal{G}_L(5,2)$, define the following distribution:

$$(p, G, G_t, e) \xleftarrow{R} \mathcal{G}_L(5,2), t', r' \xleftarrow{R} \mathbb{Z}_p$$
$$\mathbf{g}_1, \mathbf{h}_1 \xleftarrow{R} G_1, \mathbf{g}_2, \mathbf{P}_2, \mathbf{Q}_2, \mathbf{R}_2, \mathbf{S}_2 \xleftarrow{R} G_2, \mathbf{P}_3 \xleftarrow{R} G_3,$$
$$\mathbf{P}_4, \mathbf{Q}_4, \mathbf{R}_4, \mathbf{S}_4 \xleftarrow{R} G_4.$$

Then, we choose:

$$D = \left(p, G, G_t, e, \mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_1^{t'}\mathbf{R}_2, \mathbf{h}_1^{t'}\mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4, \mathbf{h}_1\mathbf{Q}_4, \mathbf{g}_1^{r'}\mathbf{S}_2\mathbf{S}_4\right),$$
$$X_1 = \mathbf{h}_1^{r'}\mathbf{Q}_2\mathbf{R}_4, \mathbf{X}_2 \xleftarrow{R} \langle G_1, G_2, G_4 \rangle.$$

The advantage for an algorithm $\mathcal{A}$ to distinguish $\mathbf{X}_1$ and $\mathbf{X}_2$ is defined:

$$\mathrm{Adv4}_{\mathcal{G}_L(5,2),\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(D, \mathbf{X}_1) = 1] - \Pr[\mathcal{A}(D, \mathbf{X}_2) = 1]|.$$

If $\mathrm{Adv4}_{\mathcal{G}_L(5,2),\mathcal{A}}(\lambda)$ is negligible, the group generator $\mathcal{G}_L(5,2)$ satisfies Assumption 4.

4.3 Proof in detail

In this subsection, we first introduced the core theorem for security proof. Then, we define the structure of secret key and ciphertext in security proof. Finally, we constructed a series of games and proved the indistinguishability between these games through six lemmas.

Theorem *If Assumptions 1 to 4 holds, then our CP-ABE scheme can be proved adaptively secure in the standard model.*

Proof We use subgroup G_2, which is not used in the normal CP-ABE construction, to help prove the security. □

Firstly, we generate the semi-function ciphertext. We first choose $y, y' \in_R \mathbb{Z}_p$ and $\mathbf{w}, \mathbf{w}' \in_R \mathbb{Z}_p^n$ randomly. Then, we choose three random numbers $z_i \in_R \mathbb{Z}_p$ related to attributes and $\alpha_i, \alpha_i' \in_R \mathbb{Z}_p$ related to the row of the matrix A in access policy. Then construct the semi-function ciphertext:

$$\mathrm{CT}_{\mathbb{A}} = \left((A, \rho), C_1, C_1', D_{1,j}, D_{1,j}', C_2, C_2', D_{2,j}\right),$$

where

$$C_1 = MY^s, C_2 = Y^{s^{'}}$$
$$C_1^{'} = \mathbf{g}_1^s \mathbf{g}_2^y, C_2^{'} = \mathbf{g}_1^{s^{'}} \mathbf{X}_2 \mathbf{g}_2^{y^{'}}, D_{1,j}^{'} = \mathbf{g}_1^{r_j} \mathbf{X}_{1,j}^{'} \mathbf{g}_2^{-\alpha_j}$$
$$D_{1,j} = \mathbf{g}_1^{bA_j \cdot \mathbf{v}_1} \left(\mathbf{g}_1^{t_{\rho(j)}} \mathbf{Z}\right)^{-r_j} \mathbf{X}_{1,j} \mathbf{g}_2^{A_j \cdot \mathbf{w} + \alpha_j z_{\rho(j)}}$$
$$D_{2,j} = \mathbf{g}_1^{bA_j \cdot \mathbf{v}_2} \left(\mathbf{g}_1^{t_{\rho_j}} \mathbf{Z}\right)^{-s^{'}} \mathbf{X}_{2,j} \mathbf{g}_2^{A_j \mathbf{w}^{'} + \alpha_j^{'} z_{\rho(j)}}.$$

Secondly, we hope to generate three types of semi-function keys. We choose $d, d^{'} \in_R \mathbb{Z}_p$ and $\{d_i \in_R \mathbb{Z}_p\}_{i \in I_S}$ randomly. Then, set three types of semi-function keys:

1. $semi{-}key_1 = (\theta, K = \mathbf{g}_1^a \mathbf{g}_1^{br} \mathbf{R}_3 \mathbf{g}_2^d, K^{'} = \mathbf{g}_1^r \mathbf{R}_3^{'} \mathbf{g}_2^{d^{'}},$
$\{K_i = \left(\mathbf{g}_1^{s_i} \mathbf{h}_1\right)^r \mathbf{R}_{3,i} \mathbf{g}_2^{d^{'} z_i}\}_{i \in I_S})$;
2. $semi{-}key_2 = (\theta, K = \mathbf{g}_1^a \mathbf{g}_1^{br} \mathbf{R}_3 \mathbf{g}_2^d, K^{'} = \mathbf{g}_1^r \mathbf{R}_3^{'},$
$\{K_i = \left(\mathbf{g}_1^{s_i} \mathbf{h}_1\right)^r \mathbf{R}_{3,i}\}_{i \in I_S})$;
3. $semi{-}key_3 = (\theta, K = \mathbf{g}_1^a \mathbf{g}_1^{br} \mathbf{R}_3 \mathbf{g}_2^d, K^{'} = \mathbf{g}_1^r \mathbf{R}_3^{'} \mathbf{g}_2^{d^{'}},$
$\{K_i = \left(\mathbf{g}_1^{s_i} \mathbf{h}_1\right)^r \mathbf{R}_{3,i} \mathbf{g}_2^{d_i}\}_{i \in I_S})$;

We use these semi-function keys and ciphertext; we can construct a list of games. We define q to be the maximum number of key queries, and $q \geq k \geq 1$. Then, we can construct the sequence games as follows:

1. $Game_{Real}$: In this game, both the secret keys queried by the adversary and the ciphertext are the same as the normal secret keys in our CP-ABE scheme.
2. $Game_{0,3}$: In this game, the secret keys queried by the adversary are the same as the normal secret keys in the above scheme. Set the challenge ciphertext to be the semi-functional ciphertext.
3. $Game_{k,1}$: In this game, the first $k-1$ secret keys queried by the adversary are $semi{-}key_3$. The k th secret key is $semi{-}key_1$. The rest secret keys are the same as the normal secret keys in the above scheme. Set the challenge ciphertext to be the semi-functional ciphertext.
4. $Game_{k,2}$: In this game, the first $k-1$ secret keys queried by the adversary are $semi{-}key_3$. Set the kth secret key to be $semi{-}key_2$. The rest secret keys are the same as the normal secret keys in the above scheme. Set the challenge ciphertext to be the semi-functional ciphertext.
5. $Game_{k,3}$: In this game, the first k secret keys queried by the adversary are $semi{-}key_3$. The rest secret keys are the same as the normal secret keys in the above scheme. Set the challenge ciphertext to be the semi-functional ciphertext.
6. $Game_{Final_0}$: In this game, all queried secret keys are $semi{-}key_3$. Set the challenge ciphertext to be semi-function encryption of a random message which is independent of M_0 and M_1.
7. $Game_{Final_1}$: This game is similar to $Game_{Final_0}$. The only difference is that $D_{1,j}$ and $D_{2,j}$ are random elements in $G_1 \times G_2 \times G_4$. Set the challenge ciphertext to be independent of attribute sets T_0 and T_1. Hence, the advantage of adversary is 0.

Finally, we propose 6 lemmas to connect above games. The target is to prove $Game_{Real}$ and $Game_{Final_1}$ are indistinguishable, so **Theorem** holds; then, our scheme is secure.

Lemma 2 *Based on Assumption 1, $Game_{Real}$ and $Game_{0,3}$ are computationally indistinguishable.*

Proof Suppose there exists an adversary $\mathcal{A}$ satisfying $|Game_{Real}\ Adv_{\mathcal{A}} - Game_{0,3} Adv_{\mathcal{A}}| = \epsilon$. We can construct a simulator $\mathcal{B}$ with $Adv1_{\mathcal{G}_{L(5,2),\mathcal{B}}}(\lambda) = \epsilon$ to break Assumption 1. $\mathcal{B}$ is given $\mathbf{g}_1, \mathbf{g}_3, \mathbf{g}_4, \mathbf{V}$ and simulates $Game_{Real}$ or $Game_{0,3}$. □

Setup $\mathcal{B}$ randomly chooses $a, b, a_0 \in \mathbb{Z}_p$ and $\mathbf{Z}_4 \in G_4$. Then, $\mathcal{B}$ sets $Y = e(\mathbf{g}_1, \mathbf{g}_1)^a, \mathbf{h}_1 = \mathbf{g}_1^{a_0}, Z = \mathbf{h}_1 \mathbf{Z}_4$ and sends $PK = \left(p, \mathbf{g}_1, \mathbf{g}_1^b, Y, \mathbf{Z}, \mathbf{g}_4\right)$ to $\mathcal{A}$.

Phase 1 $\mathcal{B}$ generates secret keys which are the same as the secret key generated in our CP-ABE scheme from $MK = (a, \mathbf{h}_1, \mathbf{g}_3)$ and can answer the key queries from $\mathcal{A}$.

Challenge $\mathcal{A}$ submits two messages M_0, M_1 of equal length and two access structures $\mathbb{A}_0 = (A, \rho, T_0), \mathbb{A}_1 = (A, \rho, T_1)$ where $\mathbb{A}_0, \mathbb{A}_1$ cannot be satisfied by any attribute set queried in phase 1. $\mathcal{B}$ randomly chooses $\beta \in \{0, 1\}$ and does:

1. Choose three vectors $\mathbf{v} = (1, v_2, \ldots, v_n) \in_R \mathbb{Z}_p^n, \mathbf{v}^{'} = \left(1, v_2^{'}, \ldots, v_n^{'}\right) \in_R \mathbb{Z}_p^n, \mathbf{v}_\delta = \left(0, v_{\delta,2}, \ldots, v_{\delta,n}\right) \in_R \mathbb{Z}_p^n$.
2. Choose $\hat{r}_j \in_R \mathbb{Z}_p, \hat{s}_j = \left(a_0 + t_{\rho(j)}\right)^{-1}$ and $\hat{X}_{1,j}, \hat{X}_{1,j}^{'}, \mathbf{X}_2, \hat{X}_{2,j} \in_R G_4$, for $1 \leq j \leq \ell$.
3. Choose $\hat{s} \in_R \mathbb{Z}_p$ and for $1 \leq j \leq \ell$ calculate:

$$C_1 = M_\beta e\left(\mathbf{g}_1^a, \mathbf{V}\right), C_1^{'} = \mathbf{V}, D_{1,j}^{'} = \mathbf{V}^{\hat{r}_j} \mathbf{X}_{1,j}^{'}$$
$$D_{1,j} = \mathbf{V}^{bA_j \cdot \mathbf{v}} \mathbf{V}^{-(a_0 + t_{\rho(j)})\hat{r}_j} \hat{X}_{1,j}$$
$$C_2 = e\left(\mathbf{g}_1, \mathbf{V}^{\hat{s}}\right), C_2^{'} = \mathbf{V}^{\hat{s}} \mathbf{X}_2$$
$$D_{2,j} = \mathbf{V}^{\hat{s} a A_j \cdot \mathbf{v}^{'}} \mathbf{V}^{((A_j \cdot \mathbf{v}_\delta) b \hat{s}_j - \hat{s})(a_0 + t_{(j)})} \hat{X}_{2,j}.$$

4. Set the challenge ciphertext as $CT_{\mathbb{A}_\beta}$:

$$CT_{\mathbb{A}_\beta} = \left((A, \rho), C_1, C_1^{'}, D_{1,j}^{'}, D_{1,j}, C_2, C_2^{'}, D_{2,j}\right).$$

If $\mathbf{V} \leftarrow \langle G_1, G_2 \rangle$, let $\mathbf{V} = \mathbf{g}_1^s \mathbf{g}_2^y$, then :

$$C_2 = Y^{s'}, C_2' = \mathbf{g}_1^{s'} \mathbf{X}_2 \mathbf{g}_2^{y'},$$
$$D_{2,j} = \mathbf{g}_1^{aA_j \cdot \mathbf{v}_2} \left(\mathbf{g}_1^{t_{\rho(j)}} Z\right)^{-s'} \mathbf{X}_{2,j} \mathbf{g}_2^{A_j \cdot \mathbf{w}' + \alpha_j' z_j},$$

where $s' = s\hat{s}$, $y' = y\hat{s}$, $\mathbf{v}_2 = \left(s\hat{s}\right)\mathbf{v}' + s\mathbf{v}_\delta$

The first component of vector $\mathbf{v}_2$ is s'. Besides, $z_{\rho(j)} = a_0 + t_{\rho(j)}, \mathbf{X}_{2,j} = \mathbf{Z}_4^{s'} \hat{X}_{2,j}, \mathbf{w}' = y\hat{s}b\mathbf{v}', \alpha_j' = y\left(\left(A_j \cdot \mathbf{v}_\delta\right) b\, \hat{s}_j - \hat{s}\right)$. We have:

$$C_1 = M_\beta Y^s, C_1' = \mathbf{g}_1^s \mathbf{g}_2^y$$
$$D_{1,j} = \mathbf{g}_1^{bA_j \cdot \mathbf{v}} \left(\mathbf{g}_1^{t_{\rho(j)}} \mathbf{Z}\right)^{-r_j} \mathbf{X}_{1,j} \mathbf{g}_2^{A_j \cdot \mathbf{w} + \alpha_j' z_{\rho(j)}}$$
$$D_{1,j}' = \mathbf{V}^{\hat{r}_j} \mathbf{X}_{1,j}' = \mathbf{g}_1^{r_j} \mathbf{X}_{1,j}' \mathbf{g}_2^{-\alpha_j},$$

where $\mathbf{v}_1 = s\mathbf{v}, r_j = s\hat{r}_j, \mathbf{X}_{1,j} = \mathbf{Z}_4^{sr_j} \hat{X_{1,j}}, z_{\rho(j)} = a_0 + t_{\rho(j)}, \alpha_j = -y\hat{r}_j$, and s is the first component of $\mathbf{v}_1$. Therefore, $\mathcal{B}$ simulates $Game_{0,3}$ for the challenge ciphertext is semi-functional.

If $\mathbf{V} \leftarrow G_1$, $\mathcal{B}$ simulates $Game_{Real}$. *Phase 2* $\mathcal{B}$ does the same operation as

Phase 1 with the restriction that the queried attribute sets cannot satisfy $\mathbb{A}_0$ and $\mathbb{A}_1$. When $\mathbf{V} \leftarrow \langle G_1, G_2 \rangle$, $\mathcal{B}$ simulates $Game_{0,3}$. When $\mathbf{V} \leftarrow G_1$, $\mathcal{B}$ simulates $Game_{Real}$. Then, $\mathbf{V}$ can be distinguished by $\mathcal{B}$ with the advantage $Adv1_{\mathcal{G}_L(5,2),\mathcal{A}}(\lambda) = \epsilon$.

Lemma 3 *Based on Assumption 2, $Game_{k-1,3}$ and $Game_{k,1}$ are computationally indistinguishable.*

Proof Suppose there exists an adversary $\mathcal{A}$ satisfying

$$|Game_{k-1,3} Adv_{\mathcal{A}} - Game_{k,1} Adv_{\mathcal{A}}| = \epsilon.$$

We can construct a simulator $\mathcal{B}$ with $Adv2_{\mathcal{G}_L(5,2),\mathcal{A}}(\lambda) = \epsilon$ to break Assumption 2. Given $\mathbf{g}_1, \mathbf{P}_1\mathbf{P}_2, \mathbf{Q}_2\mathbf{Q}_3, \mathbf{P}_3, \mathbf{P}_4, \mathbf{V}$, $\mathcal{B}$ can simulate $Game_{k-1,3}$ or $Game_{k,1}$.

Setup: $\mathcal{B}$ randomly picks $a, b, a_0 \in_R \mathbb{Z}_p$ and $\mathbf{Z}_4 \in_R G_4$. Then, it sets $Y = e(\mathbf{g}_1, \mathbf{g}_1)^a, \mathbf{h_1} = \mathbf{g}_1^{a_0}, \mathbf{Z} = \mathbf{h}_1\mathbf{Z}_4$. $\mathcal{B}$ sends $PK = \left(p, \mathbf{g}_1, \mathbf{g}_1^b, Y, \mathbf{Z}, \mathbf{g}_4\right)$, and only $\mathcal{B}$ knows $MK = (a, \mathbf{h}_1, \mathbf{g}_3)$. □

Phase 1: To answer the key queries from $\mathcal{A}$ for $\theta = (I_S, S)$ with $S = \{s_i\}_{i \in I_S}$, $\mathcal{B}$ does the following:

1. For $j < k$, choose $r, \hat{d}, \hat{d}' \in_R \mathbb{Z}_p$ and $\{\hat{d}_i \in_R \mathbb{Z}_p\}_{i \in I_S}$, then $\mathcal{B}$ can create $semi\text{-}key_3$ as follows:

$$K = \mathbf{g}_1^a \mathbf{g}_1^{br} \left(\mathbf{Q}_2\mathbf{Q}_3\right)^{\hat{d}}$$
$$K' = \mathbf{g}_1^r \left(\mathbf{Q}_2\mathbf{Q}_3\right)^{\hat{d}'}, \{K_i = \left(\mathbf{g}_1^{s_i}\mathbf{h}_1\right)^r \left(\mathbf{Q}_2\mathbf{Q}_3\right)^{\hat{d}_i}\}_{i \in I_S}.$$

2. For $j > k$, $\mathcal{B}$ can generate normal keys by using the key generation algorithm.
3. For $j = k$ (the kth query), $\mathcal{B}$ chooses $\mathbf{R}_1, \mathbf{R}, \mathbf{R}', \mathbf{R}_i \in_R G_3$ and calculate $K = \mathbf{g}_1^a \mathbf{V}^b \mathbf{R}, K' = \mathbf{V}\mathbf{R}', \{K_i = \mathbf{V}^{a_0 + s_i}\mathbf{R}_i\}_{i \in I_S}$. We observe that:

   (a) If $\mathbf{V} \leftarrow \langle G_1, G_2, G_3 \rangle$, let $\mathbf{V} = \mathbf{g}_1^r \mathbf{g}_2^{\hat{d}'} \mathbf{R}_1$, then :

$$K = \mathbf{g}_1^a \mathbf{g}_1^{br} \mathbf{R}_3 \mathbf{g}_2^d, K' = \mathbf{g}_1^r \mathbf{R}_3' \mathbf{g}_2^{d'},$$
$$\{K_i = \left(\mathbf{g}_1^{s_i}\mathbf{h}_1\right)^r \mathbf{R}_{3,i} \mathbf{g}_2^{d' z_i}\},$$

   where $\mathbf{R}_3 = \mathbf{R}_1^b \mathbf{R}, d = bd', \mathbf{R}_3' = \mathbf{R}_1\mathbf{R}', \mathbf{R}_{3,i} = \mathbf{R}_1^{a_0 + s_i}\mathbf{R}_i$. It is $semi\text{-}key_1$.

   (b) If $\mathbf{V} \leftarrow \langle G_1, G_3 \rangle$, it is a properly distributed normal key.

Challenge: Two equal-length messages M_0, M_1 with two access policies $\mathbb{A}_0 = (A, \rho, T_0)$ and $\mathbb{A}_1 = (A, \rho, T_1)$ are generated by $\mathcal{A}$ and sent to $\mathcal{B}$, where $\mathbb{A}_0, \mathbb{A}_1$ cannot be satisfied by any attribute set queried in phase 1. $\mathcal{B}$ randomly picks $\beta \in \{0, 1\}$ and does:

1. $\mathcal{B}$ creates $\mathbf{v} = (1, v_2, \ldots, v_n), \mathbf{v}' = \left(1, v_2', \ldots, v_n'\right) \in \mathbb{Z}_p^n$ and $\mathbf{v}_\delta = \left(0, v_{\delta,2}, \ldots, v_{\delta,n}\right) \in \mathbb{Z}_p^n$.
2. Choose $\hat{r}_j \in_R \mathbb{Z}_p, \hat{s}_j = \left(a_0 + t_{\rho(j)}\right)^{-1}$ and $\hat{X}_{1,j}, \mathbf{X}_{1,j}', \mathbf{X}_2, \hat{X}_{2,j} \in_R G_4$, for $1 \le j \le \ell$.
3. Choose $\hat{s} \in_R \mathbb{Z}_p$ and for $1 \le j \le \ell$ calculate:

$$C_1 = M_\beta e\left(\mathbf{g}_1^a, (\mathbf{P}_1\mathbf{P}_2)\right), C_1' = (\mathbf{P}_1\mathbf{P}_2),$$
$$D_{1,j}' = (\mathbf{P}_1\mathbf{P}_2)^{\hat{r}_j} \mathbf{X}_{1,j}'$$
$$D_{1,j} = (\mathbf{P}_1\mathbf{P}_2)^{bA_j \cdot \mathbf{v}} (\mathbf{P}_1\mathbf{P}_2)^{-(a_0 + t_{\rho(j)})\hat{r}_j} \hat{X}_{1,j}$$
$$C_2 = e\left(\mathbf{g}_1, (\mathbf{P}_1\mathbf{P}_2)^{\hat{s}}\right), C_2' = (\mathbf{P}_1\mathbf{P}_2)^{\hat{s}} \mathbf{X}_2$$
$$D_{2,j} = (\mathbf{P}_1\mathbf{P}_2)^{((A_j \cdot \mathbf{v}_\delta) b\hat{s}_j - \hat{s})(a_0 + t_{(j)})} \hat{X}_{2,j}$$
$$(\mathbf{P}_1\mathbf{P}_2)^{\hat{s} a A_j \cdot \mathbf{v}'}.$$

4. Set the challenge ciphertext as $CT_{\mathbb{A}_\beta}$:

$$CT_{\mathbb{A}_\beta} = \left((A, \rho), C_1, C_1', D_{1,j}', D_{1,j}, C_2, C_2', D_{2,j}\right).$$

Suppose $P_1P_2 = \mathbf{g}_1^s \mathbf{g}_2^y$, then:

$$C_2 = Y^{s'}, C_2' = \mathbf{g}_1^{s'} \mathbf{X}_2 \mathbf{g}_2^{y'}$$
$$D_{2,j} = \mathbf{g}_1^{aA_j \cdot \mathbf{v}_2} \left(\mathbf{g}_1^{t_{\rho(j)}} \mathbf{Z}\right)^{-s'} \mathbf{X}_{2,j} \mathbf{g}_2^{A_j \cdot \mathbf{w}' + \alpha_j' z_j},$$

where $s' = s\hat{s}$, $y' = y\hat{s}$, $\mathbf{v}_2 = \left(s\hat{s}\right)\mathbf{v}' + s\mathbf{v}_\delta$, $z_{\rho(j)} = a_0 + t_{\rho(j)}$.

Table 2 Performance comparisons of previous related work

Schemes	Large universe	Adaptive security	Decryption test	Expressiveness	Group order	Standard model
Yang et al. (2016)	×	✓	×	AND gate	Composite order	✓
Zhang et al. (2013)	×	×	✓	AND gate	Prime order	×
Zhang et al. (2019)	×	×	×	Policy tree	Prime order	✓
Zhao et al. (2019)	×	✓	✓	Policy tree	Composite order	×
Hao et al. (2019)	×	×	✓	LSSS	Prime order	×
Han et al. (2018)	✓	×	✓	LSSS	Prime order	×
Cui et al. (2018)	✓	×	×	LSSS	Prime order	×
Lai et al. (2012)	×	✓	✓	LSSS	Composite order	✓
Zhang et al. (2018)	✓	✓	✓	LSSS	Composite order	✓
Ours	✓	✓	✓	LSSS	Prime order	✓

Table 3 Computation time in composite order group and Prime order group

Group order	Exponentiation in G_1 (ms)	Exponentiation in G_t (ms)	Pairing (ms)
Composite	25.79	2.68	37.51
Prime	1.40	0.14	3.71

The first component of vector $\mathbf{v}_1^{'}$ is $s^{'}$. Besides, $\mathbf{X}_{2,j} = \mathbf{Z}_4^{s^{'}} \hat{X}_{2,j}, \mathbf{w}^{'} = y\hat{s}b\mathbf{v}^{'}, \alpha_j^{'} = y\left(\left(A_j \cdot \mathbf{v}_\delta\right) b\hat{s}_j - \hat{s}\right)$. We have:

$$C_1 = M_\beta Y^s, C_1^{'} = \mathbf{g}_1^s \mathbf{g}_2^y$$

$$D_{1,j} = \mathbf{g}_1^{bA_j \cdot \mathbf{v}} \left(\mathbf{g}_1^{t_{\rho(j)}} \mathbf{Z}\right)^{-r_j} \mathbf{X}_{1,j} \mathbf{g}_2^{A_j \cdot \mathbf{w} + \alpha_j^{'} z_{\rho(j)}}$$

$$D_{1,j}^{'} = (\mathbf{P}_1\mathbf{P}_2)^{\hat{r}_j} \mathbf{X}_{1,j}^{'} = \mathbf{g}_1^{r_j} \mathbf{X}_{1,j}^{'} \mathbf{g}_2^{-\alpha_j},$$

where $\mathbf{v}_1 = s\mathbf{v}, r_j = s\hat{r}_j, \mathbf{X}_{1,j} = \mathbf{Z}_4^{sr_j} \hat{X}_{1,j}, z_{\rho(j)} = a_0 + t_{\rho(j)}, \mathbf{w} = yb\mathbf{v}, \alpha_j = -y\hat{r}_j$, and s is the first component of $\mathbf{v}_1$. Therefore, the challenge ciphertext is semi-functional.

Phase 2: $\mathcal{B}$ works the same as **Phase 1** under a different restriction that all of the queried attribute sets cannot satisfy $\mathbb{A}_0$ and $\mathbb{A}_1$. If $\mathbf{V} \leftarrow \langle G_1, G_2, G_3\rangle$, $\mathcal{B}$ simulates $Game_{k,1}$. If $\mathbf{V} \leftarrow \langle G_1, G_3\rangle$, $\mathcal{B}$ simulates $Game_{k-1,3}$. Then, $\mathbf{V}$ can be distinguished by $\mathcal{B}$ with the advantage $Adv2_{\mathcal{G}_L(5,2),\mathcal{A}}(\lambda) = \epsilon$.

Lemma 4 *Based on Assumption 2, $Game_{k,1}$ and $Game_{k,2}$ are computationally indistinguishable.*

Proof Suppose there exists an adversary $\mathcal{A}$ satisfying

$$|Game_{k,1}Adv_{\mathcal{A}} - Game_{k,2}Adv_{\mathcal{A}}| = \epsilon.$$

□

We can construct a simulator $\mathcal{B}$ with $Adv2_{\mathcal{G}_L(5,2),\mathcal{A}}(\lambda) = \epsilon$ to break Assumption 2. Given $\mathbf{g}_1, \mathbf{P}_1\mathbf{P}_2, \mathbf{Q}_2\mathbf{Q}_3, \mathbf{P}_3, \mathbf{P}_4, \mathbf{V}$, $\mathcal{B}$ can simulate $Game_{k,1}$ or $Game_{k,2}$.

Setup: $\mathcal{B}$ randomly chooses $a, b, a_0 \in_R \mathbb{Z}_p$ and $\mathbf{Z}_4 \in_R G_4$. Set $Y = e(\mathbf{g}_1, \mathbf{g}_1)^a, \mathbf{h}_1 = \mathbf{g}_1^{a_0}, \mathbf{Z} = \mathbf{h}_1\mathbf{Z}_4$, and send $\mathcal{A}$ the $PK = \left(p, \mathbf{g}_1, \mathbf{g}_1^b, Y, \mathbf{Z}, \mathbf{g}_4\right)$. Only $\mathcal{B}$ knows $MK = (a, \mathbf{h}_1, \mathbf{g}_3)$.

Phase 1: To answer the jth key query where $j \neq k$, $\mathcal{B}$ does the same as Proof of Lemma 3.

To answer the jth key query where $j = k$, $\mathcal{B}$ does the same operations like Proof of Lemma 3, but randomly picks $e \in_R \mathbb{Z}_p$ and sets $K = \mathbf{g}_1^a \mathbf{V}^b \mathbf{R} (\mathbf{Q}_2\mathbf{Q}_3)^e, K^{'} = \mathbf{V}\mathbf{R}^{'}, \{K_i = \mathbf{V}^{a_0+s_i}\mathbf{R}_i\}_{i\in I_S}$. Here $(\mathbf{Q}_2\mathbf{Q}_3)^e$ term was added to randomize the G_2 part of K, which is the only difference between this proof and Proof of Lemma 3. If $\mathbf{V} \leftarrow \langle G_1, G_2, G_3\rangle$, this is a properly distributed $semi\text{-}key_1$. If $\mathbf{V} \leftarrow \langle G_1, G_3\rangle$, this is a properly distributed $semi\text{-}key_2$.

Challenge: The same as **Challenge** in Proof of Lemma 3.
Phase 2: $\mathcal{B}$ works the same as **Phase 1** under a different restriction that all of the queried attribute sets cannot satisfy $\mathbb{A}_0$ and $\mathbb{A}_1$. If $\mathbf{V} \leftarrow \langle G_1, G_2, G_3\rangle$, $\mathcal{B}$ simulates $Game_{k,1}$. If $\mathbf{V} \leftarrow \langle G_1, G_3\rangle$, $\mathcal{B}$ simulates $Game_{k,2}$. Then, $\mathbf{V}$ can be distinguished by $\mathcal{B}$ with the advantage $Adv2_{\mathcal{G}_L(5,2),\mathcal{A}}(\lambda) = \epsilon$.

Lemma 5 *Based on Assumption 2, $Game_{k,2}$ and $Game_{k,3}$ are computationally indistinguishable.*

Proof Suppose there exists an adversary $\mathcal{A}$ satisfying

$$|Game_{k,2}Adv_{\mathcal{A}} - Game_{k,3}Adv_{\mathcal{A}}| = \epsilon.$$

We can construct a simulator $\mathcal{B}$ with $Adv2_{\mathcal{G}_L(5,2),\mathcal{A}}(\lambda) = \epsilon$ to break Assumption 2. Given $\mathbf{g}_1, \mathbf{P}_1\mathbf{P}_2, \mathbf{Q}_2\mathbf{Q}_3, \mathbf{P}_3, \mathbf{P}_4, \mathbf{V}$, $\mathcal{B}$ can simulate $Game_{k,2}$ or $Game_{k,3}$. □

Setup: $\mathcal{B}$ randomly chooses $a, b, a_0 \in_R \mathbb{Z}_p$ and $\mathbf{Z}_4 \in_R G_4$. Set $Y = e(\mathbf{g}_1, \mathbf{g}_1)^a$, $\mathbf{h}_1 = \mathbf{g}_1^{a_0}$, $\mathbf{Z} = \mathbf{h}_1\mathbf{Z}_4$, and send $\mathcal{A}$ the $PK = \left(p, \mathbf{g}_1, \mathbf{g}_1^b, Y, \mathbf{Z}, \mathbf{g}_4\right)$.

Phase 1: To answer the jth key query where $j \neq k$, $\mathcal{B}$ does the same as Proof of Lemma 3.

To answer the jth key query where $j = k$, $\mathcal{B}$ chooses $f, e \in_R \mathbb{Z}_p, \mathbf{R}, \mathbf{R}', \mathbf{R}_i \in_R G_3$ and calculates:

$$K = \mathbf{g}_1^a \mathbf{V}^{fb} \mathbf{R} \left(\mathbf{Q}_2\mathbf{Q}_3\right)^e, K' = \mathbf{V}\mathbf{R}',$$
$$\{K_i = \mathbf{V}^{(a_0+s_i)f}\mathbf{R}_i\}_{i\in I_S}.$$

If $\mathbf{V} \leftarrow \langle G_1, G_2, G_3\rangle$, let $\mathbf{V} = \mathbf{g}_1^{r'}\mathbf{g}_2^{\hat{d}}\mathbf{R}_1$ where $r' \leftarrow_R \mathbb{Z}_p$, then

$$K = \mathbf{g}_1^a \mathbf{g}_1^{br}\mathbf{R}_3\mathbf{g}_2^d, K' = \mathbf{g}_1^r\mathbf{R}_3'\mathbf{g}_2^{d'},$$
$$\{K_i = \left(\mathbf{g}_1^{s_i}\mathbf{h}_1\right)^r \mathbf{R}_{3,i}\mathbf{g}_2^{d_i}\}_{i\in I_S}.$$

where $r = fr'$, $\mathbf{g}_2^d = \mathbf{g}_2^{bf\hat{d}}\mathbf{Q}_2^e$, $\mathbf{R}_3 = \mathbf{R}_1^{bf}\mathbf{R}_1\mathbf{Q}_3^e$, $\mathbf{R}_3' = \mathbf{R}_1^f\mathbf{R}'$, $d' = f\hat{d}$, $\mathbf{R}_{3,i} = \mathbf{R}_1^{f(a_0+s_i)}\mathbf{R}_i$, $d_i = f(a_0+s_i)\hat{d}_i$. This is a properly distributed $semi\text{-}key_3$. If $\mathbf{V} \leftarrow \langle G_1, G_3\rangle$, this is a properly distributed $semi\text{-}key_2$.

Challenge: The same as **Challenge** in Proof of Lemma 3.

Phase 2: $\mathcal{B}$ works the same as **Phase 1** under a different restriction that all of the queried attribute sets cannot satisfy $\mathbb{A}_0$ and $\mathbb{A}_1$. If $\mathbf{V} \leftarrow \langle G_1, G_2, G_3\rangle$, $\mathcal{B}$ simulates $Game_{k,3}$. If $\mathbf{V} \leftarrow \langle G_1, G_3\rangle$, $\mathcal{B}$ simulates $Game_{k,2}$. Then, $\mathbf{V}$ can be distinguished by $\mathcal{B}$ with the advantage $Adv2_{\mathcal{G}_{L(5,2)},\mathcal{A}}(\lambda) = \epsilon$.

Lemma 6 *Based on Assumption 3, $Game_{q,3}$ and $Game_{Final_0}$ are computationally indistinguishable.*

Proof Suppose there exists an adversary $\mathcal{A}$ satisfying

$$|Game_{q,3}Adv_{\mathcal{A}} - Game_{Final_0}Adv_{\mathcal{A}}| = \epsilon.$$

We can construct a simulator $\mathcal{B}$ with $Adv3_{\mathcal{G}_{L(5,2)},\mathcal{A}}(\lambda) = \epsilon$ to break Assumption 3. Given $\mathbf{g}_1$, $\mathbf{g}_2$, $\mathbf{g}_1^a\mathbf{P}_2$, $\mathbf{g}_1^s\mathbf{Q}_2$, $\mathbf{P}_3$, $\mathbf{P}_4$, V, $\mathcal{B}$ can simulate $Game_{q,3}$ or $Game_{Final_0}$. □

Setup: $\mathcal{B}$ randomly chooses $b, a_0 \in_R \mathbb{Z}_p$ and $\mathbf{Z}_4 \in_R G_4$. Then $\mathcal{B}$ sets $Y = e\left(\mathbf{g}_1, \mathbf{g}_1^a\mathbf{P}_2\right)$, $\mathbf{h}_1 = \mathbf{g}_1^{a_0}$, $\mathbf{Z} = \mathbf{h}_1\mathbf{Z}_4$, and send $\mathcal{A}$ $PK = \left(p, \mathbf{g}_1, \mathbf{g}_1^b, Y, \mathbf{Z}, \mathbf{g}_4\right)$.

Phase 1: To answer the key queries and the normal keys for $\theta = (I_S, S)$ with $S = \{s_i\}_{i\in I_S}$, $\mathcal{B}$ chooses $r, \hat{d}, d' \in_R \mathbb{Z}_p, \{d_i \in_R \mathbb{Z}_p\}_{i\in I_S}$, and $\mathbf{R}_3, \mathbf{R}_3', \mathbf{R}_{3,i} \in_R G_3$, then creates $semi\text{-}key_3$:

$$K = \left(\mathbf{g}_1^a\mathbf{P}_2\right)\mathbf{g}_1^{br}\mathbf{R}_3\mathbf{g}_2^{\hat{d}} = \mathbf{g}_1^a\mathbf{g}_1^{br}\mathbf{R}_3\mathbf{g}_2^d,$$
$$K' = \mathbf{g}_1^r\mathbf{R}_3'\mathbf{g}_2^{d'}, \{K_i = (\mathbf{g}_1\mathbf{h}_1)^r\mathbf{R}_{3,i}\mathbf{g}_2^{d_i}\}_{i\in I_S}.$$

where $\mathbf{g}_2^d = \mathbf{P}_2\mathbf{g}_2^{\hat{d}}$.

Table 4 Performance comparisons of previous related work

Schems	Size of ciphertext	Encryption cost	Decryption test cost	Decryption cost
Lai et al. (2012)	$(4\ell+2)\mathbb{G}+2\mathbb{G}_t$	$(8\lvert I\rvert+2)\mathrm{Exp}_{\mathbb{G}}+2\mathrm{Exp}_{\mathbb{G}_t}$	$\lvert I\rvert\mathrm{Exp}_{\mathbb{G}_t}+\lvert 2I+1\rvert\mathrm{Pair}_c$	$\lvert I\rvert\mathrm{Exp}_{\mathbb{G}_t}+(2\lvert I\rvert+1)\mathrm{Pair}_c$
Zhang et al. (2018)	$(3\ell+2)\mathbb{G}+2\mathbb{G}_t$	$(6\lvert I\rvert+3)\mathrm{Exp}_{\mathbb{G}}+2\mathrm{Exp}_{\mathbb{G}_t}$	$2\lvert I\rvert\mathrm{Exp}_{\mathbb{G}}+2\mathrm{Pair}_c$	$\lvert I\rvert\mathrm{Exp}_{\mathbb{G}_t}+(2\lvert I\rvert+1)\mathrm{Pair}_c$
Ours	$(15\ell+10)G+2G_t$	$(30\lvert I\rvert+15)\mathrm{Exp}_G+2\mathrm{Exp}_{G_t}$	$10\lvert I\rvert\mathrm{Exp}_G+10\mathrm{Pair}_p$	$\lvert I\rvert\mathrm{Exp}_{G_t}+(10\lvert I\rvert+5)\mathrm{Pair}_p$

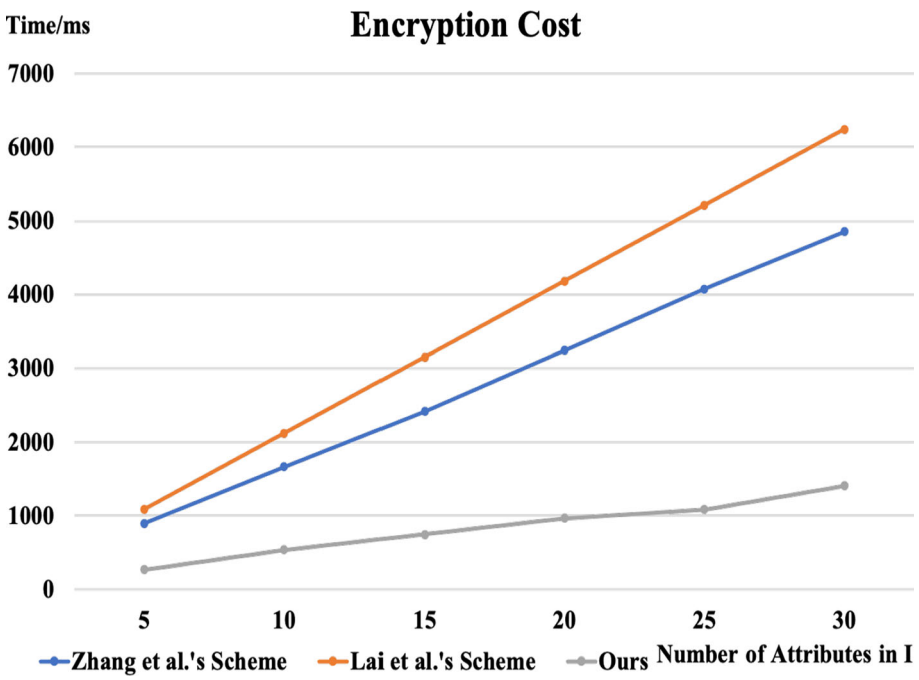

Fig. 3 Encryption cost

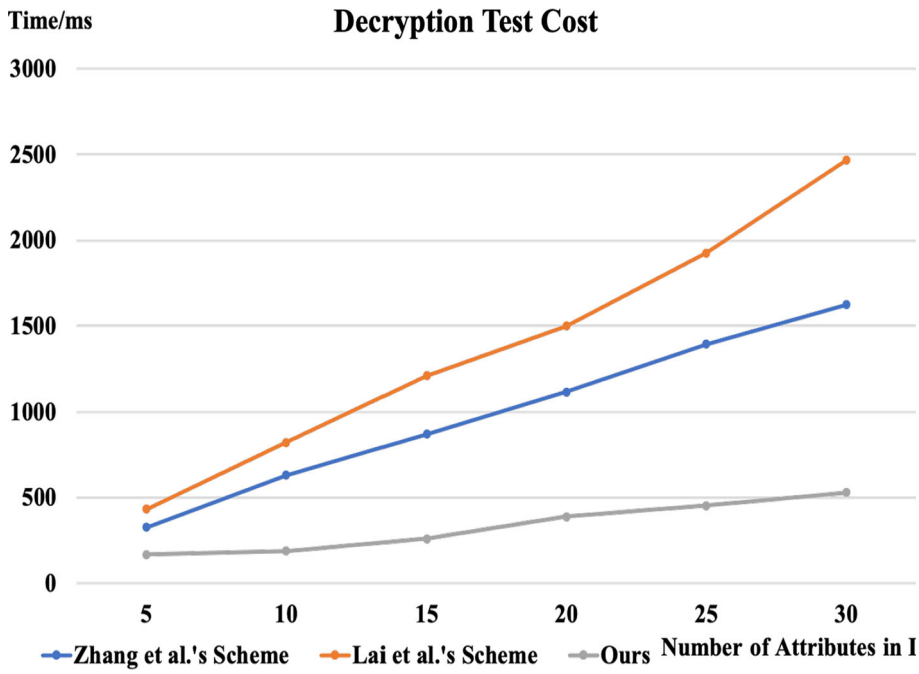

Fig. 4 Decryption test cost

Challenge: Two equal-length messages M_0, M_1 with two access policies $\mathbb{A}_0 = (A, \rho, T_0)$ and $\mathbb{A}_1 = (A, \rho, T_1)$ are generated by $\mathcal{A}$ and sent to $\mathcal{B}$, where $\mathbb{A}_0, \mathbb{A}_1$ cannot be satisfied by any attribute set queried in phase 1. $\mathcal{B}$ randomly picks $\beta \in \{0, 1\}$ and does:

1. $\mathcal{B}$ creates $\mathbf{v} = (1, v_2, \ldots, v_n) \in_R \mathbb{Z}_p$ for $2 \leq i \leq n$, and chooses $\mathbf{v}_1^{'} = (s^{'}, v_{2,2}, \ldots, v_{2,n})$, $\mathbf{w}^{'} = (w_1^{'}, w_2^{'}, \ldots, w_n^{'} \in_R \mathbb{Z}_p)$.
2. $\mathcal{B}$ chooses $\hat{r}_j, \alpha_j^{'} \in_R \mathbb{Z}_p$ and $\mathbf{X}_2, \mathbf{X}_{2,j}, \hat{X}_{1,j}, X_{1,j}^{'} \in_R G_4$ for $1 \leq j \leq \ell$.
3. Let $T_\beta = \left(t_{\rho(1)}, t_{\rho(2)}, \ldots, t_{\rho(\ell)}\right)$. $\mathcal{B}$ chooses $y^{'} \in \mathbb{Z}_p$ and then calculate:

$$C_1 = M_\beta V, C_1^{'} = \mathbf{g}_1^s \mathbf{Q}_2,$$
$$D_{1,j} = \left(\mathbf{g}_1^s \mathbf{Q}_2\right)^{bA_j \cdot \mathbf{v}} \left(\mathbf{g}_1^s \mathbf{Q}_2\right)^{-(a_0 + t_{\rho(j)})\hat{r}_j} \hat{X}_{1,j},$$
$$D_{1,j}^{'} = \left(\mathbf{g}_1^s \mathbf{Q}_2\right)^{\hat{r}_j} X_{1,j}^{'},$$

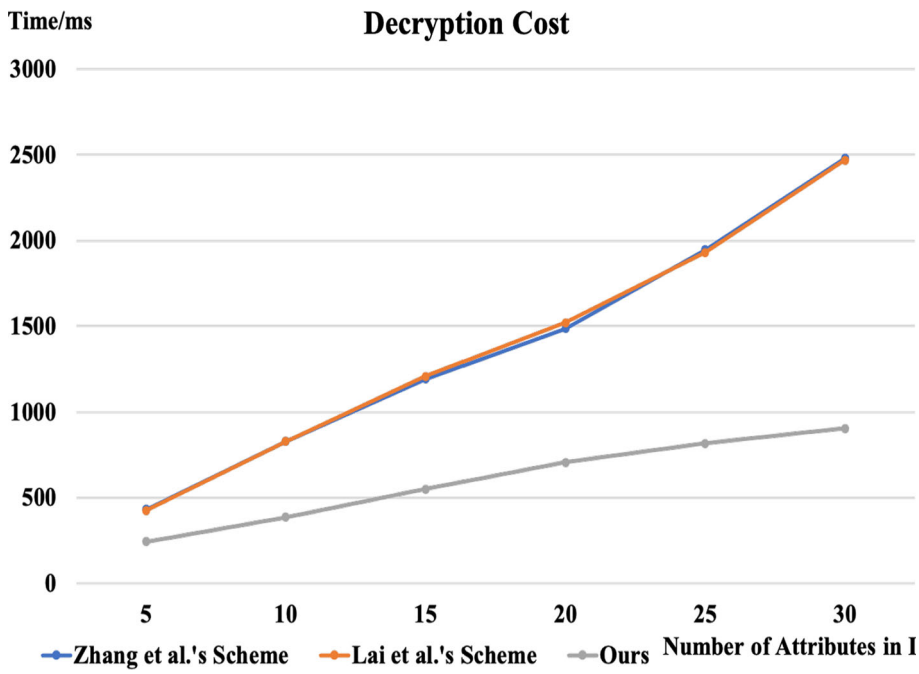

Fig. 5 Decryption cost

$$C_2 = Y^{s'}, C_2' = \mathbf{g}_1^{s'} \mathbf{X}_2 \mathbf{g}_2^{y'},$$
$$D_{2,j} = \mathbf{g}_1^{bA_j \cdot \mathbf{v}'} \left(\mathbf{g}_1^{t_{\rho(j)}} Z\right)^{-s'} \mathbf{X}_{2,j} \mathbf{g}_2^{A_j \cdot \mathbf{w}' + \alpha_j' (a_0 + t_{\rho(j)})},$$

where $1 \le j \le \ell$.

4. $\mathcal{B}$ sends the challenge ciphertext $CT_{\mathbb{A}_\odot}$ to $\mathcal{A}$.

$$CT_{\mathbb{A}_\beta} = \left((A, \rho), C_1, C_1', D_{1,j}, D_{1,j}', C_2, C_2', D_{2,j}\right).$$

If $\mathbf{g}_1^s \mathbf{Q}_2 = \mathbf{g}_1^s \mathbf{g}_2^y$, then we have the challenge ciphertext as follows:

$$C_1 = M_\beta V, C_1' = \mathbf{g}_1^s \mathbf{g}_2^y,$$
$$D_{1,j} = \mathbf{g}_1^{bA_j \cdot \mathbf{v}_1} \left(\mathbf{g}_1^{t_{\rho(j)}} \mathbf{Z}\right)^{-r_j} \mathbf{X}_{1,j} \mathbf{g}_2^{A_j \cdot \mathbf{w} + \alpha_j z_{\rho(j)}},$$
$$D_{1,j}' = \left(\mathbf{g}_1^s Y_2\right)^{\hat{r}_j} \mathbf{X}_{1,j}' = \mathbf{g}_1^{r_j} \mathbf{X}_{1,j}' \mathbf{g}_2^{-\alpha_j'},$$
$$C_2 = Y^{s'}, C_2' = \mathbf{g}_1^{s'} \mathbf{X}_2 \mathbf{g}_2^{y'},$$
$$D_{2,j} = \mathbf{g}_1^{bA_j \cdot \mathbf{v}_1'} \left(\mathbf{g}_1^{t_{\rho(j)}} Z\right)^{-s'} \mathbf{X}_{2,j} \mathbf{g}_2^{A_j \cdot \mathbf{w}' + \alpha_j' z_{\rho(j)}},$$

where $\mathbf{v}_1 = s\mathbf{v}, r_j = s\hat{r}_j, \mathbf{X}_{1,j} = \mathbf{Z}_4^{r_j} \hat{X}_{1,j}, \mathbf{w} = yb\mathbf{v}$, $\alpha_j = -y\hat{r}_j, z_{\rho(j)} = a_0 + t_{\rho(j)}$.

Phase 2: $\mathcal{B}$ works the same as **Phase 1** under a different restriction that all of the queried attribute sets cannot satisfy $\mathbb{A}_0$ and $\mathbb{A}_1$. If $T = e(\mathbf{g}_1, \mathbf{g}_1)^{as}$, the distribution of challenge ciphertext is identical to the distribution of semi-functional encryption of M_β, so $\mathcal{B}$ simulates $Game_{q,3}$. Otherwise, the distribution of challenge ciphertext is identical to the distribution of semi-functional encryption of a random message in G_t, so $\mathcal{B}$ simulates $Game_{Final_0}$. Then, T can be distinguished by $\mathcal{B}$ with the advantage $Adv3_{\mathcal{G}_L(5,2),\mathcal{A}}(\lambda) = \epsilon$.

Lemma 7 *Based on Assumption 4, $Game_{Final_0}$ and $Game_{Final_1}$ are computationally indistinguishable.*

Proof Suppose there exits an adversary $\mathcal{A}$ satisfying

$$|Game_{Final_0} Adv_{\mathcal{A}} - Game_{Final_1} Adv_{\mathcal{A}}| = \epsilon.$$

We can construct a simulator $\mathcal{B}$ with $Adv4_{\mathcal{G}_L(5,2),\mathcal{A}}(\lambda) = \epsilon$ to break Assumption 4. Given $\mathbf{g}_1, \mathbf{g}_2, \mathbf{R}_2 \mathbf{g}_1^{t'}, \mathbf{P}_2 \mathbf{h}_1^{t'}, \mathbf{P}_3, \mathbf{P}_4, \mathbf{h}_1 \mathbf{Z}_4, \mathbf{g}_1^{r'} \mathbf{S}_2 \mathbf{S}_4, \mathbf{V}$, $\mathcal{B}$ can simulate $Game_{Final_0}$ or $Game_{Final_1}$. □

Setup: $\mathcal{B}$ randomly chooses $a, b \in_R \mathbb{Z}_p$. Then set $Y = e(\mathbf{g}_1, \mathbf{g}_1)^a, \mathbf{Z} = \mathbf{h}_1 \mathbf{Z}_4$, and sends $\mathcal{A}$ $PK = (p, \mathbf{g}_1, \mathbf{g}_1^b, Y, \mathbf{Z}, \mathbf{g}_4)$.

Phase 1: When $\mathcal{A}$ asks for a key for $\theta = (I_S, S)$ with $S = \{s_i\}_{i \in I_S}$, $\mathcal{B}$ randomly picks $\hat{t} \in_R \mathbb{Z}_p$, and $\mathbf{R}_{3,i}, \mathbf{R}_3, \mathbf{R}_3' \in_R G_3$ for $i \in I_S$, then set $semi\text{-}key_3$ as follows:

$$K = \left(\mathbf{g}_1^a \mathbf{g}_1^{t'} \mathbf{R}_2\right)^{b\hat{t}} \mathbf{R}_3, K' = \left(\mathbf{g}_1^{y'} \mathbf{R}_2\right)^{\hat{t}} \mathbf{R}_3',$$
$$\{K_i = \left(\mathbf{g}_1^{t'} \mathbf{R}_2\right)^{s_i \hat{t}} \left(\mathbf{h}_1^{t'} \mathbf{P}_2\right)^{\hat{t}} \mathbf{R}_{3,i}\}_{i \in I_S}.$$

In fact, $K = \mathbf{g}_1^a \mathbf{g}_1^{bt} \mathbf{R}_3 \mathbf{g}_2^d$, $\{K_i = \left(\mathbf{g}_1^{s_i} \mathbf{h}_1\right)^t \mathbf{R}_{3,i} \mathbf{g}_2^{d_i}\}_{i \in I_S}$, $K' = \mathbf{g}_1 \mathbf{R}_3' \mathbf{g}_2^{d'}$, where $t = t'\hat{t}, \mathbf{g}_2^d = \mathbf{R}_2^{b\hat{t}}, \mathbf{g}_2^{d'} = \mathbf{R}_2^{\hat{t}}, \mathbf{g}_2^{d_i} = \mathbf{R}_2^{s_i \hat{t}} \mathbf{P}_2^{\hat{t}}$.

It is a properly distributed *semi-key*$_3$.

Challenge: Two equal-length messages M_0, M_1 with two access policies $\mathbb{A}_0 = (A, \rho, T_0)$ and $\mathbb{A}_1 = (A, \rho, T_1)$ are generated by $\mathcal{A}$ and sent to $\mathcal{B}$, where $\mathbb{A}_0, \mathbb{A}_1$ cannot be satisfied by any attribute set queried in phase 1. $\mathcal{B}$ randomly picks $\beta \in \{0, 1\}$ and does:

1. Choose $\mathbf{v}_1 = \left(s, v_{1,2}, \ldots, v_{1,n}\right)$, $\mathbf{v}^{'} = (s^{'}, v_2^{'}, \ldots, v_n^{'})$, $\mathbf{v}_\delta = \left(0, v_{\delta,2}, \ldots, v_{\delta,n}\right)$, and $\mathbf{w}, \mathbf{w}^{'} \in_R \mathbb{Z}_p^n$.
2. Choose $\hat{r}_j \in_R \mathbb{Z}_p, \hat{s}_j = t_{\rho(j)}^{-1}, \mathbf{X}_2, \hat{X}_{2,j}, \hat{X}_{1,j} \in_R G_4$, for $1 \le j \le \ell$.
3. Choose $y, y^{'} \in \mathbb{Z}_p$ and then calculate:

$$C_1 =\leftarrow G_t, C_1^{'} = \mathbf{g}_1^s \mathbf{g}_2^y,$$

$$D_{1,j} = \mathbf{g}_1^{bA_j \cdot \mathbf{v}_1} \left(\mathbf{g}_1^{r^{'}} \mathbf{S}_2 \mathbf{S}_4\right)^{-\hat{r}_j t_{\rho(j)}} \mathbf{V}^{-\hat{r}_j} \mathbf{g}_2^{A_j \cdot \mathbf{w}} \hat{X}_{1,j}$$

$$D_{1,j}^{'} = \left(\mathbf{g}_1^{r^{'}} \mathbf{S}_2 \mathbf{S}_4\right)^{\hat{r}_j}$$

$$C_2 = Y^{s^{'}}, C_2^{'} = \mathbf{g}_1^{s^{'}} \mathbf{X}_2 \mathbf{g}_2^{y^{'}}$$

$$D_{2,j} = \mathbf{g}_1^{bA_j \cdot \mathbf{v}^{'}} \left(\mathbf{g}_1^{r^{'}} \mathbf{S}_2 \mathbf{S}_4\right)^{A_j \cdot \mathbf{v}_\delta b} \mathbf{V}^{b\hat{s}_j (A_j \cdot \mathbf{v}_\delta t_{\rho(j)})}$$

$$\left(\mathbf{g}_1^{t_{\rho(j)}} \mathbf{Z}\right)^{-s^{'}} \hat{X}_{2,j} \mathbf{g}_2^{A_j \cdot \mathbf{w}^{'}}.$$

4. $\mathcal{B}$ sends the challenge ciphertext to $\mathcal{A}$:

$$CT_{\mathbb{A}_\beta} = \left((A, \rho), C_1, C_1^{'}, D_{1,j}, D_{1,j}^{'}, C_2, C_2^{'}, D_{2,j}\right).$$

If $\mathbf{V} = \mathbf{h}_1^{r^{'}} \mathbf{Q}_2 \mathbf{R}_4$, suppose $\mathbf{h}_1 = \mathbf{g}_1^{\tau_1}, \mathbf{S}_2 = \mathbf{g}_2^{\gamma}, \mathbf{Q}_2 = \mathbf{g}_2^{\gamma \tau_2}$ with $\tau_1, \tau_2, \gamma \in \mathbb{Z}_p$. Then $C_2 = Y^{s^{'}}, C_2^{'} = \mathbf{g}_1^{s^{'}} \mathbf{X}_2 \mathbf{g}^{y^{'}}, C_{2,j} = \mathbf{g}_1^{bA_j \cdot \mathbf{v}_1^{'}} \left(\mathbf{g}_1^{t_{\rho(j)}} \mathbf{Z}\right)^{-s^{'}} \mathbf{g}_2^{A_j \cdot \mathbf{w}^{'} + \alpha_j^{'} z_{\rho(j)}}$ $\mathbf{X}_{2,j}$, where $\mathbf{v}_1^{'} = \mathbf{v}^{'} + \mathbf{v}_\delta \left(r^{'} + r^{'} \tau_1\right)$.

We have that the first component of $\mathbf{v}_1^{'}$ is $s^{'}$, and $\alpha_j^{'} = \left(A_j \cdot \mathbf{v}_\delta \gamma \tau_2 b \hat{s}_j \left(1 - \tau_2 \left(t_{\rho(j)+\tau_2}\right)^{-1}\right)\right), z_{\rho(j)} = \tau_2 + t_{\rho(j)}, \mathbf{X}_{2,j} = \mathbf{S}_4^{A_j \cdot \mathbf{v}_\delta b} \mathbf{R}_4^{A_j \cdot \mathbf{v}_\delta b} \hat{X}_{2,j}$. Besides, we have

$$C_1 \leftarrow_R G_t, C_1^{'} = \mathbf{g}_1^s \mathbf{g}_2^y,$$

$$D_{1,j} = \mathbf{g}_1^{bA_j \cdot \mathbf{v}_1} \left(\mathbf{g}_1^{t_{\rho(j)}} \mathbf{Z}\right)^{-r_j} \mathbf{g}_2^{A_j \cdot \mathbf{w} + \alpha_j z_{\rho(j)}} \mathbf{X}_{1,j},$$

$$D_{1,j}^{'} = \left(\mathbf{g}_1^{r^{'}} \mathbf{S}_2 \mathbf{S}_4\right)^{\hat{r}_j} = \mathbf{g}_1^{r_j} \mathbf{X}_{1,j}^{'} \mathbf{g}_2^{-\alpha_j},$$

where $r_j = r^{'} \hat{r}_j, \alpha_j = -\gamma \hat{r}_j, z_{\rho(j)} = \tau_2 + t_{\rho(j)}$, $\mathbf{X}_{1,j} = \mathbf{S}_4^{-\hat{r}_j t_{\rho(j)}} \mathbf{R}_4^{-\hat{r}_j} \mathbf{Z}_4^{r^{'} \hat{r}_j} \hat{X}_{1,j}, \mathbf{X}_{1,j}^{'} = \mathbf{S}_4^{\hat{r}_j}$.

Phase 2: $\mathcal{B}$ works the same as **Phase 1** under a different restriction that all of the queried attribute sets cannot satisfy $\mathbb{A}_0$ and $\mathbb{A}_1$. If $T = \mathbf{h}_1^{r^{'}} \mathbf{Q}_2 \mathbf{R}_4$, the distribution of challenge ciphertext is identical to the distribution of the semi-functional encryption of a random message in G_t, so $\mathcal{B}$ simulates $Game_{Final_0}$. Otherwise, if $T \leftarrow < G_1, G_2, G_4 >$, $\mathcal{B}$ simulates $Game_{Final_1}$. Then, T can be distinguished by $\mathcal{B}$ with the advantage $Adv4_{\mathcal{G}_{L(5,2)},\mathcal{A}}(\lambda) = \epsilon$.

4.4 Analysis of security for VFC

In this section, we analysis the security of our system.

1. *Privacy of Plaintext* According to the proof of our CP-ABE scheme, the adversary cannot recover the plaintext, even they can get the secret keys related to some attribute sets. Therefore, our system can protect the privacy of ciphertext;
2. *Collusion Resistance* In the **KeyGen** phase of our CP-ABE scheme, the K_i are associated with every attribute. There are different random values r when generating K_i for different users. Therefore, users cannot simply combine their attributes to generate a secret key that can decrypt the ciphertext, unless at least one of them has the secret key which can decrypt the ciphertext;
3. *Partially Policy Hiding* In our CP-ABE scheme, only the attribute names are contained in the access structure. Adversary cannot get any information about attribute values from the access structure. Therefore, the proposed system can only expose attribute names that are not sensitive and hide the sensitive attribute values;

5 Performance analysis

In this section, we will compare our scheme with previous works.

In Table 2, we compare some important features of previous policy hiding CP-ABE schemes and ours. The schemes of Zhao et al. (2019), Zhang et al. (2019), Hao et al. (2019), Yang et al. (2016), Zhang et al. (2013), and Lai et al. (2012) do not support the large universe. The schemes of Han et al. (2018) and Cui et al. (2018) support the large universe, but they are only selectively secure, while our scheme is adaptively secure. The schemes of Zhang et al. (2019), Yang et al. (2016), and Cui et al. (2018) do not have a decryption test in the decryption step. The schemes of Zhao et al. (2019), Han et al. (2018), Hao et al. (2019), Zhang et al. (2013), and Cui et al. (2018) are proved security in the random oracle, while our scheme is proved security in the standard model. The scheme of Zhang et al. (2018) has a decryption test and can be proved adaptively secure in the standard model, but

this scheme is inefficient because they use composite order groups as the basic group.

We further compare our scheme with the schemes of Lai et al. (2012) and Zhang et al. (2018). Both Lai et al. (2012) and Zhang et al. (2018) use the composite order groups that are very inefficient compared with prime order groups. According to the analysis of De Caro and Iovino (2011), to satisfy the security level equivalent to 1024 bit discrete logarithm security, we should choose prime order groups with 512 bits' elements or 4-primes composite order groups of 1024 bits' elements. (Elements in every prime order subgroup are 256 bits.) We test the time of exponentiation in G_1, the time of exponentiation in G_t, and time of pairing on a laptop (with 1.4GHz Intel i5-8257U CPU, and 16GB RAM) based on macOS Big Sur 11.0.1 and Java pairing-based cryptography library 2.0.0 (De Caro and Iovino (2011)). We choose Type A1 pairings and Type A pairings, which are both built on the curve $y^2 = x^3 + x$. In Table 3, we show the comparison of calculating time between the composite order group and prime order group. Then we compare our scheme with the schemes of Lai et al. (2012) and Zhang et al. (2018) in detail in Table 4. The definitions of the notations in the table are as follows:

- $\mathbb{G}$: is a composite order group of order N which is the product of 4 prime numbers and every prime is 256 bits. The elements in $\mathbb{G}$ and its subgroups are 1024 bits in length;
- G: is a prime order group of order p, where p is a 160-bit prime, and the elements in G are 512 bits.;
- ℓ: denotes the number of rows in the matrix A in access policy;
- $|I|$: denotes the number of minimum authorized attribute set;
- $\text{Exp}_{\mathbb{G}}$: denotes the time of exponentiation in $\mathbb{G}$;
- $\text{Exp}_{\mathbb{G}_t}$: denotes the time of exponentiation in $\mathbb{G}_t$;
- Exp_G: denotes the time of exponentiation in G;
- Exp_{G_t}: denotes the time of exponentiation in G_t;
- Pair_c: denotes the time of pairing in composite order group $\mathbb{G}$;
- Pair_p: denotes the time of pairing in prime order group G;

Although the size of ciphertext in our scheme is larger than the schemes of Zhang et al. (2018) and Lai et al. (2012), the computation in encryption, decryption test, and decryption is faster than both schemes. The comparison details of encryption, decryption test, and decryption are shown in Figs. 3, 4, 5. In Fig. 3, we show that our scheme is about 3.4 times faster than the scheme of Zhang et al. (2018) in the encryption phase. In Fig. 4, we show that our scheme is about 3 times faster than the scheme of Zhang et al. (2018) in the decryption test phase, and in Fig. 5 we show that ours is about 2 times faster in the decryption phase.

In summary, our scheme is more efficient than existing schemes in computation cost. Besides, we prove that our scheme is adaptively secure in the standard model and can partially hide attributes in access policy with efficient decryption test before decryption. Therefore, our scheme is suitable for the smart transportation environment.

6 Conclusion

We summarized the overall architecture of VFC and the security requirements of VFC. Then, we proposed a CP-ABE scheme based on the prime order bilinear pairing group for the security requirements of VFC. After that, we proved its adaptive security under the standard model. Finally, a performance analysis was made.

However, there are also some open problems. The size of ciphertext is too large, which is not suitable for storage limited devices. How to reduce the length of the ciphertext is an open problem. Besides, the pairing operation in the decryption phase will grow with the minimum number of attributes required in the access policy, which is not efficient enough. How to reduce the number of pairing operations in the decryption phase to a constant level is also worth studying.

Acknowledgements This work was supported in part by the Sichuan Science and Technology Program (2019YFG0506 and 2020YFG0292).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Alrawais A, Alhothaily A, Hu C, Xing X, Cheng X (2017) An attribute-based encryption scheme to secure fog communications. IEEE Access 5:9131–9138

Bethencourt J, Sahai A, Waters B (2007) Ciphertext-policy attribute-based encryption. In: 2007 IEEE symposium on security and privacy (SP'07). IEEE, pp 321–334

Chen H, Liao Y (2019) Improvement of an outsourced attribute-based encryption scheme. Soft Comput 23(22):11409–11417. https://doi.org/10.1007/s00500-019-04088-y

Cui H, Deng RH, Lai J, Yi X, Nepal S (2018) An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures, revisited. Comput Netw 133:157–165

De Caro A, Iovino V (2011) jpbc: Java pairing based cryptography. In: 2011 IEEE symposium on computers and communications (ISCC). IEEE, pp 850–855

Feng C, Yu K, Aloqaily M, Alazab M, Lv Z, Mumtaz S (2020) Attribute-based encryption with parallel outsourced decryption for edge intelligent iov. IEEE Trans Veh Technol 69(11):13784–13795

Freeman DM (2010) Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In: Annual international conference on the theory and applications of cryptographic techniques. Springer, pp 44–61

Goyal V, Pandey O, Sahai A, Waters B (2006) Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM conference on computer and communications security. ACM, pp 89–98

Guillevic A (2013) Comparing the pairing efficiency over composite-order and prime-order elliptic curves. In: International conference on applied cryptography and network security. Springer, pp 357–372

Han Q, Zhang Y, Li H (2018) Efficient and robust attribute-based encryption supporting access policy hiding in internet of things. Future Gener Comput Syst 83:269–277

Hao J, Huang C, Ni J, Rong H, Xian M, Shen XS (2019) Fine-grained data access control with attribute-hiding policy for cloud-based IoT. Comput Netw 153:1–10

Hou X, Li Y, Chen M, Wu D, Jin D, Chen S (2016) Vehicular fog computing: a viewpoint of vehicles as the infrastructures. IEEE Trans Veh Technol 65(6):3860–3873

Huang C, Lu R, Choo KKR (2017) Vehicular fog computing: architecture, use case, and security and forensic challenges. IEEE Commun Mag 55(11):105–111

Jiang Y, Susilo W, Mu Y, Guo F (2018) Ciphertext-policy attribute-based encryption against key-delegation abuse in fog computing. Future Gener Comput Syst 78:720–729

Katz J, Sahai A, Waters B (2008) Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Annual international conference on the theory and applications of cryptographic techniques. Springer, pp 146–162

Lai J, Deng RH, Li Y (2011) Fully secure cipertext-policy hiding cp-abe. In: International conference on information security practice and experience. Springer, pp 24–39

Lai J, Deng RH, Li Y (2012) Expressive cp-abe with partially hidden access structures. In: Proceedings of the 7th ACM symposium on information, computer and communications security. ACM, pp 18–19

Lee CC, Chung PS, Hwang MS (2013) A survey on attribute-based encryption schemes of access control in cloud environments. Int J Netw Secur 15(4):231–240

Lee CC, Lai YM, Cheng PJ (2016) An efficient multiple session key establishment scheme for vanet group integration. IEEE Intell Syst 31(6):35–43. https://doi.org/10.1109/IVS.2015.7225898

Liao Y, Zhang G, Chen H (2020) Cost-efficient outsourced decryption of attribute-based encryption schemes for both users and cloud server in green cloud computing. IEEE Access 8:20862–20869. https://doi.org/10.1109/ACCESS.2020.2969223

Ning Z, Huang J, Wang X (2019) Vehicular fog computing: enabling real-time traffic management for smart cities. IEEE Wirel Commun 26(1):87–93

Nishide T, Yoneyama K, Ohta K (2008) Attribute-based encryption with partially hidden encryptor-specified access structures. In: International conference on applied cryptography and network security. Springer, pp 111–129

Sahai A, Waters B (2005) Fuzzy identity-based encryption. In: Annual international conference on the theory and applications of cryptographic techniques. Springer, pp 457–473

Tian H, Li X, Quan H, Chang CC, Baker T (2020) A lightweight attribute-based access control scheme for intelligent transportation system with full privacy protection. IEEE Sensors J. https://doi.org/10.1109/JSEN.2020.3030688

Xiao Y, Zhu C (2017) Vehicular fog computing: vision and challenges. In: 2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops). IEEE, pp 6–9

Yang K, Han Q, Li H, Zheng K, Su Z, Shen X (2016) An efficient and fine-grained big data access control scheme with privacy-preserving policy. IEEE Internet Things J 4(2):563–571

Zhang Y, Chen X, Li J, Wong DS, Li H (2013) Anonymous attribute-based encryption supporting efficient decryption test. In: Proceedings of the 8th ACM SIGSAC symposium on information, computer and communications security. ACM, pp 511–516

Zhang Y, Zheng D, Deng RH (2018) Security and privacy in smart health: efficient policy-hiding attribute-based access control. IEEE Internet of Things J 5(3):2130–2145

Zhang Y, Li J, Yan H (2019) Constant size ciphertext distributed cp-abe scheme with privacy protection and fully hiding access structure. IEEE Access 7:47982–47990

Zhao Y, Zhang X, Xie X, Ding Y, Kumar S (2019) A verifiable hidden policy cp-abe with decryption testing scheme and its application in vanet. Trans Emerg Telecommun Technol. https://doi.org/10.1002/ett.3785