Journal of Network Intelligence ©2026 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **11**, Number **1**, February **2026**

A Fine-Grained Access Control Scheme for Multi-Authority under the Group IoV

Yan-ming Lu[1], Rang Zhou[1,*]

[1]College of Computer Science and Cyber Security,
Chengdu University of Technology, Chengdu, Sichuan 610059, China
298528001@qq.com, zhourang@cdut.edu.cn

Hu Xiong[2]

[2]School of Information and Software Engineering,
University of Electronic Science and Technology of China, Sichuan 610054, China
xionghu@uestc.edu.cn

*Corresponding author: Rang Zhou

Received January 21, 2025, revised June 2, 2025, accepted October 17, 2025.

Abstract. *In the Internet of Vehicle, the authorization level of vehicles is dynamically adjusted with the change of driving location, which requires the information security mechanism of the Internet of Vehicle to be able to adapt to the dynamically changing situation of vehicle location. In order to solve the access control problem caused by the change of vehicle location in different regions, this paper proposes a multi-authority attribute encryption scheme for the regional cluster Internet of Vehicle. Based on the characteristics of regional data access authorization, the scheme introduces a regional access control mechanism on the basis of the traditional distributed authorization attribute encryption scheme. Specifically, we propose an attribute hierarchical access control policy to ensure that the same attribute set is granted different access rights in different regions according to regional requirements and effectively prevent users in the current region from accessing data in other regions through illegal keys. In addition, the scheme is equipped with outsourced decryption capability, which makes it possible to further improve the system efficiency and flexibility by outsourcing the decryption operation of authority in the complex group Internet of Vehicle. At the same time, the scheme retains the decentralized feature of distributed authorization, avoiding the problem of decentralization failure that may be caused when an authorizer is in control of high-level attributes. Finally, we provide formal security proof of the scheme and verify its effectiveness through experimental simulation. The experimental results show that the scheme is able to maintain efficient performance while realizing the access control of the complex regional cluster Internet of Vehicle and meets the dual requirements of security and performance.*

Keywords: Sharing data integrity audit; Data privacy protection; Identity privacy protection; Dynamic user groups.

1. **Introduction.** Internet of Vehicle (IoV), as a kind of intelligent interconnection technology, lays a solid foundation for realizing intelligent transportation and intelligent driving by establishing information synergy between vehicles and the cloud and between vehicles and vehicles. At the same time, the continuous development of in-vehicle intelligence technology promotes the significant progress of IoV technology, especially the popularization of 5G communication technology and the continuous iteration of intelligent vehicles, which makes the design and realization of IoV more feasible and diversified

[1, 2, 3, 4, 5]. Accompanying the rapid development of IoV technology is its increasingly complex and diverse communication modes, and these interaction and collaboration processes inevitably bring about a large number of new information security challenges, such as information leakage, illegal access, and information privilege revocation.

In the field of information security under IoV, access control is a key technology that has attracted a lot of attention. By establishing the access control relationship between ciphertext and key, the information access process can be effectively protected. These public key cryptographic algorithms for realizing access control usually belong to the function encryption system [6, 7]. The complete concept of functional encryption was proposed by Dan Boneh et al. More precisely, function encryption is a public-key cryptographic algorithm that implements access control by establishing a functional relationship between the key and the ciphertext, and by mapping variables in the ciphertext during decryption. Common function encryption systems include predicate encryption systems such as identity encryption [8, 9, 10, 11, 12], attribute encryption [13, 14, 15, 16, 17, 18], and inner product encryption [19, 20, 21]. Among these predicate encryption structures, attribute encryption is one of the most popular and widely studied due to its flexibility and malleability. Attribute encryption is capable of realizing access control schemes with varying functionality through multiple algorithm designs [22, 23, 24, 25, 26, 27, 28]. Among the various attribute encryption algorithms, multi-authority attribute encryption is considered to be the most suitable scheme for information security protection in the Internet of Vehicles [29, 30, 31, 32]. This is because the authorization of keys in the multi-authority model no longer relies on a single authorization center, but is done by multiple independent authorized bodies. This decentralized authorization structure not only reduces the security risks that may be caused by centralized authority, but also distributes the authorization bodies according to the dynamic location of the vehicle and avoids the problem of information interaction caused by vehicle movement.

Despite the above advantages of the ABE algorithm with multiple authorties, its access control protection mechanism has certain shortcomings in the vehicle cluster network of the Internet of Vehicles (as shown in Figure. 1).

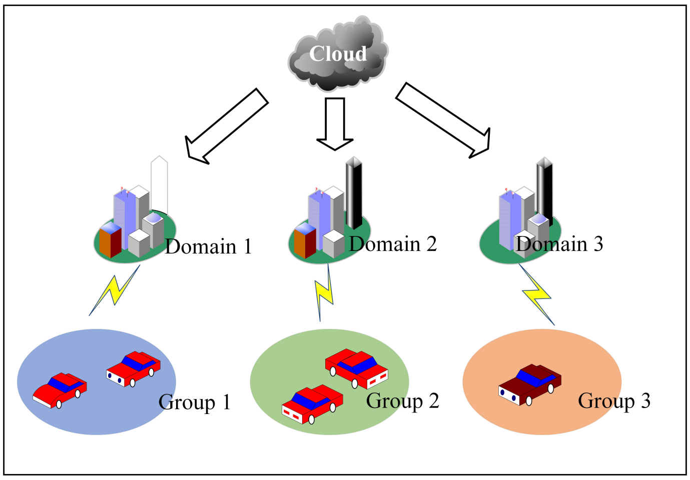

Figure 1. Group IoV

More precisely, since vehicle cluster networks are usually designed with multiple levels of permissions, unauthorized vehicles of other cluster networks should not be allowed to access the data of the current cluster, even if the same attribute set occurs in different vehicle cluster networks. This is because, in cluster vehicular networks, there are differences in attribute set permissions between different clusters, and attributes used to identify the cluster network usually have higher priority than other regular access attributes. Then for this application requirement, if we refer to the scheme in RW15 [31], we can find that the scheme does not solve this problem effectively. The fundamental reason is that the RW15 [31] scheme has a unified design for the attribution of different attributes, and all attributes are managed by different specific authorities. However, this design does not take into account the authority differences among attributes but simply arranges all attributes in a uniform manner, which makes the scheme difficult to operate effectively when facing attribute sets with significant authority differences. In addition, no matter how the high-level attributes are coordinated, as long as they are managed by any particular authority it may lead to security risks. More importantly, the existence of such high-privilege level attributes somehow undermines the original purpose of the multi-authority mechanism - i.e., the segmentation of privileges. Therefore, how to effectively circumvent the security problems caused by the differences in privileges at the algorithmic level, while maintaining the original advantages of the multi-authority mechanism and avoiding the security risks caused by a single authority, is still a key issue that needs to be solved urgently.

In order to solve the above problems, we propose a distributed authority attribute encryption scheme to support group vehicular networking based on the scheme in RW15 [31]. The scheme establishes a group network-oriented access control mechanism by embedding group identifiers in the algorithm, thus realizing fine-grained access control for group vehicular networking. In addition, the scheme also supports the revocation mechanism of group authorization: since the authority of all keys is closely related to the region where the vehicle is located after embedding the group identity when the vehicle leaves the current group network, the new key that has been given will not be able to be used for decrypting ciphertexts of the old group network. What's more, the scheme also has the capability of outsourcing decryption, which allows the decryption process to be outsourced and executed among the authorities, thus improving the flexibility and computational efficiency of the system. We also retain the characteristics of distributed authority to ensure that each authority has the same weight level, which avoids attribute privilege differentiation from destroying the balance of distributed authority. Finally, we provide formal security proof of the scheme to show that it can effectively resist CPA attacks. Through application experiments, we demonstrate the effectiveness of the scheme in group vehicular networking, which can provide more fine-grained access control protection and high operational efficiency. Our remaining sections are organized as follows: in Section II we present the background knowledge related to the scheme, in Section III we present our scheme and prove the security of the scheme, in Section IV we apply the scheme and subject it to a performance analysis, and in Section V we summarize the proposed work and give potential future directions for improvement.

2. Preliminaries.

2.1. Bilinear map.

Set G_1 and G_2 be two multiplicative cyclic groups with the same big prime order q. g_1 and g_2 are generators of the group G_1, defining $e : G_1 \times G_1 \to G_2$ is a bilinear map with the following three properties:
(1) Computability: For any $g_1, g_2 \in G_1$, there is an efficient algorithm that calculates the value of $e(g_1, g_2)$.
(2) Bilinearity: For all $g_1, g_2 \in G_1$ and $a, b \in_R Z_q^*$, $e({g_1}^a, {g_2}^b) = e(g_1, g_2)^{ab}$.
(3) Nondegeneray: The existence of $g_1, g_2 \in G_1$, $e(g_1, g_2) \neq 1$.

2.2. Difficult problem assumptions[31].

Choose a bilinear group $\mathbb{G}$ of order p according to the security parameter κ, which admits a non-degenerate bilinear mapping e: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Picks $s, a, b_1, b_2, ..., b_q \leftarrow_R \mathbb{Z}_p$ and $R \leftarrow_R \mathbb{G}$:

$$D = \left(\begin{array}{l} \mathbb{G}, p, e, g, g^s, \{g^{a^i}\}_{\substack{i\in[2q] \\ i\neq q+1}}, \{g^{b_j a^i}\}_{\substack{(i,j)\in[2q,q] \\ i\neq q+1}}, \\ \{g^{s/b_i}\}_{i\in[q]}, \{g^{sa^i b_j/b_{j'}}\}_{\substack{(i,j,j')\in[q+1,q,q] \\ j\neq j'}} \end{array} \right)$$

The assumption states that no polynomial-time distinguisher can distinguish the distribution $(D, e(g,g)^{sa^{q+1}})$ from the distribution (D, R) with more than negligible advantage.

2.3. **Access Structure.** Definition 1(Access Structures[31]): Let $P_1, P_2, ..., P_n$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1,P_2,...,P_n\}}$ is monotone if $\forall B, C$: $B \subseteq \mathbb{A}$ and $B \subseteq C$ then $C \in \mathbb{A}$. An access structure (respectively, monotone collection) is a collection (respectively, monotone collection) $\mathbb{A}$ of non-empty subsets of $\{P_1, ..., P_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{P_1,P_2,...,P_n\}} \backslash \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.

Definition 2(Linear Secret - Sharing Schemes[31]): Let p be a prime and U the attribute universe. A secret-sharing scheme Π with domain of secrets $\mathbb{Z}_p$ realizing access structures on U is linear over $\mathbb{Z}_p$ if:

- The shares of a secret $z \in \mathbb{Z}_p$ for each attribute form a vector over $\mathbb{Z}_p$.
- For each access structure $\mathbb{A}$ on U, there exists a matrix $A \in \mathbb{Z}_p^{l\times n}$, called the share-generating matrix, and a function δ, that labels the rows of A with attributes from U, i.e. $\delta : [l] \to U$, which satisfy the following: During the generation of the shares, we consider the column vector $\vec{v} = (z, r_2, ..., r_n)^{\perp}$, where $r_2, ..., r_n \leftarrow_R \mathbb{Z}_p^{l\times 1}$. Then the vector of l shares of the secret z according to Π is equal to $\vec{\lambda} = A\vec{v} \in \mathbb{Z}_p^{l\times 1}$. The share $\lambda_{j,j\in[l]}$ "belongs" to attribute $\delta(j)$. We will be referring to the pair (A, δ) as the policy of the access structure $\mathbb{A}$.

2.4. **Syntax of scheme.** $GlobalSetup(1^\lambda) \to GP$: The global setup algorithm accepts security parameters λ, and outputs the public global parameters of the system. We require that the global parameters contain a description of the attribute domain U, the permission domain U_a, the global identifier domain GID and the hash map T.

$AuthSetup(GP, a) \to \{PK_a, SK_a\}$: The authority $a \in U_a$ takes the global parameter GP as input and outputs its public key/secret key $\{PK_a, SK_a\}$.

$KeyGen(GID, id_g, u, SK_a, GP) \to K_{GID,id_g,u}$: The key generation algorithm accepts the user's global identifier GID, the group identifier id_g , the secret key of authority a, the attribute u controlled by that authority, and global parameters. It outputs a key for the identity-attribute pair (GID, id_g, u).

$KeyTrans(K_{GID,id_g,S}, GP) \to \bar{K}_{GID,id_g,S}$: The algorithm is run by the key holder and it is used to generate the semi-keys for performing outsourced decryption.

$Encrypt(M, (A, \delta), \{PK_a\}, GP) \to CT$: The encryption algorithm accepts a message M, an access control policy (A, δ), a set of related authority public keys $\{PK_a\}$ and global parameters. The output ciphertext CT.

$OutsourceDecrypt(CT, \bar{K}_{GID,id_g,S}, GP) \to \bar{D}$: The outsourced decryption algorithm utilizes the converted key to accomplish the base decryption of the ciphertext.

$Decrypt(CT, \bar{D}, K_{GID,id_g,u}, GP) \to M$: The decryption algorithm accepts a ciphertext CT, a set of keys corresponding to individual user GIDs for different attributes u, and global parameters. It outputs the message M when the set of attributes satisfies the access structure of the ciphertext, or the decryption fails.

2.5. **Static Security**[31]. GP parameter generation: the challenger invokes the algorithm $GlobalSetup(1^{\lambda}) \rightarrow GP$ and sends the global parameter GP to the adversary.

Adversary query: The adversary provides the following:

(1) A set of attributes belonging to the malicious authority $\mathcal{C}_a \subseteq U_a$, and their corresponding public keys $\{PK_a\}_{a \in \mathcal{C}_a}$.

(2) A set of attribute sets belonging to a normal authority $\mathcal{N}_a \subseteq U_a$ that does not intersect with the malicious authority. The adversary requests the public key for this attribute set.

(3) A set of secret keys queries $Q - \{(GID_i, S_i)\}_{i=1}^{m}$, where GID_i are distinct global identities, $S \subseteq U$, and $T(S_i) \cap \mathcal{C}_a = \emptyset$. Note that (GID_i, S_i) denotes the key for which the adversary asks the user with identity GID_i to provide the set of attributes S_i. That is, for each attribute $u \in S_i$, the adversary obtains its corresponding secret key $K_{GID,id_g,u}$. According to $T(S_i) \cap \mathcal{C}_a = \emptyset$, none of such secret keys comes from a malicious authority.

(4) Two equal-length messages M_0, M_1, and the access control structure $\mathbb{A}$ used for the challenge. We require that for each $i \in [m]$, the set $S_i \cup \bigcup\limits_{a \in \mathcal{C}_a} T^{-1}(a)$ is an unauthorized set of access control structures $\mathbb{A}$, where $\bigcup\limits_{a \in \mathcal{C}_a} T^{-1}(a)$ represents the set of all attributes attributed to the malicious authority.

Challenger Response: The challenger $\mathcal{S}$ flips a coin, randomly selects $b \leftarrow_R \{0,1\}$ and responds to the above request:

(1) Run $PK_a \leftarrow AuthSetup(GP, a)$, replies with the public key of the authority.

(2) For all $i \in [m]$ and $u \in S_i$ generate secret key $K_{GID_{i,u}} \leftarrow KeyGen(GID_i, SK_{T(u)}, u, GP)$.

(3) Generate challenger ciphertext $CT^* \leftarrow Enc(M_b, A, \{PK_a\}, GP)$.

Guess: The adversary outputs a guess $b' \in \{0,1\}$.

3. **Scheme.** For convenience, before giving a detailed description of the scheme construction, we give the annotations of the relevant parameters in the scheme, as shown in Table 1 below:

Table 1. Scheme Symbol

Symbol	Define
$\mathbb{G}_T, \mathbb{G}, \mathbb{Z}_p$	$\mathbb{G}_T, \mathbb{G}$ is the bilinear group, $\mathbb{Z}_p$ is a finite field of prime order.
F_2, H	Mapping any string to a hash function in $\mathbb{G}$
U	U represents the set of all attributes in the system
U_a	U_a illustrates the division of attributes in U across different authority.
T	Mapping U to U_a
S, u	u is a element in the attribute sets S

3.1. **Detail.**

- $GlobalSetup(1^{\lambda}) \rightarrow GP$: The algorithm is used to generate global public parameters GP (Global Parameters) which are next used for input to the distributed authorization center and to generate the respective public and private key parameters. First, the algorithm generates the bilinear group based on the input system parameters $\mathbb{G}$, where its prime order is p and the generating element is g. Next, it generates the following hash functions: $H : GID \rightarrow \mathbb{G}$ for mapping the global identity , $F_2 : id_g \rightarrow \mathbb{G}$ for mapping the group identity , and note that we require that the hash mapping domain of H, F_2 has no intersection. A hash function for mapping attribute values (strings) to group elements $F_1 : \{0,1\}^n \to \mathbb{G}$. Finally, the algorithm publishes the above generated content as well as the attribute domain U, the distributed authority attribute domain U_a, the mapping relation (hash function $T(U) \to U_a$) as a global public parameter $GP = \{p, e, \mathbb{G}, \mathbb{G}_T, H, F_1, F_2, U, U_a, T\}$.

- $AuthSetup(GP, a) \to \{PK_a, SK_a\}$: Each authority institution executes its own Setup algorithm, the AuthSetup algorithm. The algorithm does this by entering the global parameter GP, the authority institution identifier a, and randomly selecting two indices: $\varphi_a, \beta_a \leftarrow_R \mathbb{Z}_p$, and finally generating and publishing the public key $PK = \{e(g,g)^{\varphi_a}, g^{\beta_a}\}$, the authority institution secret key $SK = \{\varphi_a, \beta_a\}$.

- $KeyGen(GID, id_g, u, SK_a, GP) \to K_{GID,id_g,u}$: The input to the key generation algorithm consists of any attribute u in the set of attributes S, the group identity id_g, the secret key of the authority institution SK_a, the global identity GID of the key holder, and the global parameter GP. Note that $u \in T^{-1}(a)$, i.e., the attribute is governed by the current specific authority a. The algorithm selects two random numbers $t \leftarrow_R \mathbb{Z}_p, r \leftarrow_R \mathbb{Z}_p$ and outputs the key: $K_{GID,id_g,u} = \{K_1 = g^{\varphi_a} H(GID)^{\beta_a} F_1(u)^t F_2(id_g)^r, K_2 = g^t, K_3 = g^r\}$

- $KeyTrans(K_{GID,id_g,S}, GP) \to \bar{K}_{GID,id_g,S}$: The algorithm is run by the key holder and its used to generate the half-key for performing outsourced decryption. The algorithm chooses the random number: $z_1, z_2 \leftarrow_R \mathbb{Z}_p$, and does the following computation on the key: $\bar{K}_{GID,id_g,S} = \{\bar{K}_1 = K_1 \cdot F_2(id_g)^{z_1}, \bar{K}_2 = K_2 \cdot F_2(id_g)^{z_2}, \bar{K}_3 = K_3 \cdot F_1(S)^{-z_2}\}$. Note that the random element $F_2(id_g)^{z_1}$ is retained by the key holder itself for use in the decryption process after the outsourced decryption is complete.

- $Encrypt(M, (A, \delta), \{PK_a\}, GP) \to CT$: The encryption algorithm input the message M, the access control policy (A, δ), the public key $\{PK_a\}$, and the global parameter GP. Additionally, we define: $\rho : [l] \to U_a$, i.e. $\rho(\cdot) = T(\delta(\cdot))$, to map each row (attribute) in the access control structure $\mathbb{A}$ to the corresponding authority. To implement secret sharing, the algorithm first creates vectors $\vec{v} = (z, v_2, ..., v_n)^\top$ and vectors $\vec{w} = (0, w_2, ..., w_n)^\top$, where $z, v_2, ..., v_n, w_2, ..., w_n \leftarrow_R \mathbb{Z}_p$. Denote the sharing value of each row attribute x to the secret value z by $\lambda_x = \langle A_x, \vec{v} \rangle$ and the sharing value to 0 by $w_x = \langle A_x, \vec{w} \rangle$. For each row of attribute x in A, random numbers $t_x \leftarrow_R \mathbb{Z}_p$ are chosen for it. The secret message is generated as follows:

$$CT = \left\{ \begin{array}{l} C_0 = Me(g,g)^z \\ \left\{ \begin{array}{l} C_{1,x} = e(g,g)^{\lambda_x} e(g,g)^{\varphi_{\rho(x)} t_x} \\ C_{2,x} = g^{-t_x}, \\ C_{3,x} = g^{\beta_{\rho(x)} t_x} g^{w_x}, \\ C_{4,x} = F_1(\delta(x))^{t_x}, \\ C_{5,x} = F_2(id_g)^{t_x} \} \end{array} \right\}_{x \in [l]} \end{array} \right\}$$

- $OutsourceDecrypt(CT, \bar{K}_{GID,id_g,S}, GP) \to \bar{D}$: The outsourced decryption algorithm utilizes the converted key to complete the base decryption of the ciphertext and transmits the obtained partial decryption result to the key holder:

$$\begin{aligned}\bar{D} &= e(\bar{K}_1, C_{2,x})e(\bar{K}_2, C_{4,x})e(\bar{K}_3, C_{5,x}) \\ &= e(g^{\varphi_a} H(GID)^{\beta_a} F_1(u)^t F_2(id_g)^r F_2(id_g)^{z_1}, g^{-t_x})e(g^t F_2(id_g)^{z_2}, F_1(\delta(x))^{t_x}) \\ &\quad e(g^r F_1(S)^{-z_2}, F_2(id_g)^{t_x}) \\ &= e(g,g)^{-\varphi_a t_x} e(H(GID), g)^{-\beta_x t_x} e(F_1(u), g)^{-t \cdot t_x} e(F_2(id_g), g)^{-r \cdot t_x} \\ &\quad e(g, F_1(\delta(x)))^{t \cdot t_x} e(g, F_2(id_g))^{r \cdot t_x} e(F_2(id_g)^{z_1}, g^{-t_x}) \\ &= e(g,g)^{-\varphi_a t_x} e(H(GID), g)^{-\beta_x t_x} e(F_2(id_g)^{z_1}, g^{-t_x}).\end{aligned}$$

- *Decrypt*$(CT, \bar{D}, K_{GID,id_g,u}, GP) \to M$: First, the key holder uses $F_2(id_g)^{z_1}$ to disambiguate the interfering elements in $\bar{D}$:

  $D_1 = \frac{\bar{D}}{e(F_2(id_g)^{z_1}, C_{2,x})} = e(g,g)^{-\varphi_a t_x} e(H(GID), g)^{-\beta_x t_x}$.

  Next, assuming that (A, δ) is the access control policy of the ciphertext CT, if the decryptor's key $K_{GID,id_g,S}$ causes each row vector in A to be tensored to $(1, 0, ..., 0)$, then each row vector has the following computation

  $$\begin{aligned}&D_1 \cdot C_{1,x} \cdot e(H(GID), C_{3,x}) \\ &= e(g,g)^{-\varphi_a t_x} e(H(GID), g)^{-\beta_x t_x} e(g,g)^{\lambda_x} e(g,g)^{\varphi_{\rho(x)} t_x} e(H(GID), g^{\beta_{\rho(x)} t_x} g^{w_x}) \\ &= e(g,g)^{\lambda_x} e(H(GID), g)^{w_x}\end{aligned}$$

  At this point, we reach the final step of the decryption, the secret reduction process. At this point, the decrypting party computes the constant $c_x \in \mathbb{Z}_p$ that makes $\sum c_x A_x = (1, 0, ..., 0)$ and performs the secret reduction:

  $\prod\limits_x \left(e(g,g)^{\lambda_x} e(H(GID), g)^{w_x}\right)^{c_x} = e(g,g)^z$

  Finally we get the decrypted message: $M = C_0 / e(g,g)^z$.

3.2. **Formal proof.** In this section, we prove the Indistinguishable Security (IND) of our proposed scheme by playing a Static Security game between an adversary and a challenger . It is worth noting that since the conversion key used in outsource decryption is essentially a re-randomization of the base key, we are able to prove the security of the conversion key at the same time by simply completing the generalization of the base key interaction process. To achieve the above goal, we give the following theorem:

Theorem 1: If the q-DPBDHE2 assumption holds, then for any polynomial-time running adversary $\mathcal{A}$ (with maximum query matrix size $q \times q$) has a negligible advantage in statically undermining the security of our scheme.

Proof: To prove Theorem 1, first assume that there exists a polynomial-time adversary $\mathcal{A}$, whose goal is to attack our proposed scheme. Correspondingly, there exists a challenger $\mathcal{C}$, which runs the RW15 scheme; there exists a challenger $\mathcal{J}$, which provides the q-DPBDHE2 challenge tuple; and there exists a simulator $\mathcal{S}$, which mediates between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$, receives the RW15 parameters from the challenger $\mathcal{C}$ and converts them into our scheme, which is eventually provided to the adversary $\mathcal{A}$ for use in the challenge. Next, we define static security games one by one.

GP parameter generation: The simulator $\mathcal{S}$ initially accepts the q-DPBDHE2 challenge tuple from the challenger $\mathcal{J}$ and sends it to the challenger $\mathcal{C}$. The challenger $\mathcal{C}$ generates GP parameters according to the method in RW15 and sends them to the simulator $\mathcal{S}$, which defines additional hash functions $F_2 : id_g \to \mathbb{G}$ and sends them to the challenger $\mathcal{C}$.

Adversary query: The adversary provides the following to the simulator $\mathcal{S}$:

(1) A set of attributes belonging to the malicious authority $\mathcal{C}_a \subseteq U_a$, and their corresponding public keys $\{PK_a\}_{a\in\mathcal{C}_a}$.

(2) A set of attribute sets belonging to a normal authority $\mathcal{N}_a \subseteq U_a$ that does not intersect with the malicious authority. The adversary requests the public key for this attribute set.

(3) A set of secret keys queries $Q - \{(GID_i, S_i)\}_{i=1}^{m}$, where GID_i are distinct global identities, $S \subseteq U$, and $T(S_i) \cap \mathcal{C}_a = \emptyset$. Note that (GID_i, S_i) denotes the key for which the adversary asks the user with identity GID_i to provide the set of attributes S_i. That is, for each attribute $u \in S_i$, the adversary obtains its corresponding secret key $K_{GID,id_g,u}$. According to $T(S_i) \cap \mathcal{C}_a = \emptyset$, none of such secret keys comes from a malicious authority.

(4) Two equal-length messages M_0, M_1, and the access control structure $\mathbb{A}$ used for the challenge. We require that for each $i \in [m]$, the set $S_i \cup \bigcup\limits_{a\in\mathcal{C}_a} T^{-1}(a)$ is an unauthorized set of access control structures $\mathbb{A}$, where $\bigcup\limits_{a\in\mathcal{C}_a} T^{-1}(a)$ represents the set of all attributes attributed to the malicious authority.

Challenger Response: The challenger $\mathcal{S}$ flips a coin, randomly selects $b \leftarrow_R \{0,1\}$ and responds to the above request:

(1) The simulator $\mathcal{S}$ requests the authority public key PK_a from the challenger $\mathcal{C}$, and the challenger $\mathcal{C}$ runs the authority public key algorithm in RW15 to generate the authority public key PK_a, where $a \in \mathcal{N}_a$. The simulator $\mathcal{S}$ sends the authority public key PK_a to the adversary $\mathcal{A}$.

(2) The simulator $\mathcal{S}$ requests keys from the challenger $\mathcal{C}$, where all $u \in S_i$, all $i \in [m]$. The challenger $\mathcal{C}$ runs the RW15 key generation algorithm and generates a key to be transmitted to the simulator $\mathcal{S}$, which chooses a random number $r \leftarrow_R \mathbb{Z}_p$ and generates $K_3 = g^r$, and additionally, the simulator $\mathcal{S}$ performs an operation on the secret key in RW15, obtaining

$$K_1 = K_{GID_i,u} F_2(id_g)^r = g^{\varphi_a} H(GID)^{\beta_a} F_1(u)^t F_2(id_g)^r$$

Eventually, the simulator $\mathcal{S}$ sends the secret key $K_{GID_i,id_g,u}$ to the adversary.

(3) The simulator $\mathcal{S}$ generates the ciphertext of RW15 using the public key PK_a of the authority generated by the challenger $\mathcal{C}$ and the global parameter GP, and the access control structure $\mathcal{A}$ provided by the adversary, as described in section 2.5. Then, the simulator generates an additional ciphertext $\{C_{5,x} = F_2(id_g)^{t_x}\}_{x\in[l]}$. Finally, the simulator $\mathcal{S}$ sends the challenge ciphertext CT^* to the adversary $\mathcal{A}$.

Guess: The adversary outputs a guess $b' \in \{0,1\}$.

With the above game, we generalize the adversary's attack on the scheme presented in this paper to an attack on the RW15 scheme. That is, if there exists an adversary in polynomial time who is able to undermine our scheme by a non-negligible margin, then that adversary is equally able to solve the q-DPBDHE2 difficulty assumption by a non-negligible margin. Therefore, our scheme is statically secure provided that the q-DPBDHE2 difficulty assumption holds. Proof complete.

4. **Comparison.** In this section we will apply the algorithms proposed in this paper in practice on a general-purpose platform and compare their operation efficiency with that of each algorithm in the related programs. Before that, we first compare the theoretical metrics of the related algorithms, and the related symbols used in the comparison of theoretical metrics are detailed in Table 2.

First of all, in Table 3 we give the alignment on the public key size, and it is obvious that all the schemes except LCCG21 keep the same parameters as RW15.

Table 2. Comparison Symbol

Symbol	Define
$\mathbb{G}_T, \mathbb{G}$	$\mathbb{G}_T, \mathbb{G}$ is the bilinear group.
l	Number of rows in the access control structure $\mathbb{A}$
$\lvert S \rvert$	Size of the user's attribute set
U_{auth}	Number of authorities

Table 3. Public Key Size Comparison

Related	Public Key Sizes
RW15[31]	$(\mathbb{G}_T + \mathbb{G})U_{auth}$.
KAP20[29]	$(\mathbb{G}_T + \mathbb{G})U_{auth}$.
PYTY22[32]	$(\mathbb{G}_T + \mathbb{G})U_{auth}$
LCCG21[30]	$(\mathbb{G}_T + 2\mathbb{G})U_{auth}$
Ours	$(\mathbb{G}_T + \mathbb{G})U_{auth}$

Table 4. Secret key Size Comparison

Related	Key Sizes
RW15[31]	$2\lvert S \rvert \mathbb{G}$.
KAP20[29]	$(1 + 4\lvert S \rvert)\mathbb{G}$.
PYTY22[32]	$2\lvert S \rvert \mathbb{G}$.
LCCG21[30]	$(1 + 4\lvert S \rvert)\mathbb{G}$.
Ours	$3\lvert S \rvert \mathbb{G}$.

In Table 4 we show the secret key parameters of this paper's scheme and related schemes, our scheme has S more group elements than RW15 and PYTY22, but much less than the 2S+1 group parameters of KAP20 and LCCG21.

Table 5. Ciphertext Size Comparison

Related	Ciphertext Sizes	Group Control	Privilege revocation	Outsource
RW15[31]	$(4l\mathbb{G}_T + \mathbb{G})U_{auth}$.	No	No	No
KAP20[29]	$(5l\mathbb{G}_T + \mathbb{G})U_{auth}$.	No	Strong	No
PYTY22[32]	$(4l\mathbb{G}_T + (l+1)\mathbb{G})U_{auth}$	No	No	yes
LCCG21[30]	$(4l\mathbb{G}_T + 2\mathbb{G})U_{auth}$	No	No	No
Ours	$(5l\mathbb{G}_T + \mathbb{G})U_{auth}$	Yes	Weak	No

In Table 5 we show two metrics, ciphertext size and functionality comparison. For ciphertext size, our scheme is slightly higher than RW15 and the same as KAP20. In terms of functionality, our scheme is the only scheme with group access control, and the only one (our scheme and KAP20) with privilege revocation capability. Compared with dynamic revocation for policy update, our scheme only supports key privilege revocation through region switching, i.e., a key with the same set of attributes can no longer decrypt the ciphertexts of the original region by a new key in a new region (which still has the same access control attribute set). decrypt the ciphertext of the original region.

4.1. **implement.** The experiments in this paper are based on the charm [33] implementation, which uses SS512 elliptic curves. We conduct our experiments on Ubuntu 16:04 LTS operating system on Intel(R) Core(TM) i5-1035G4CPU@1.10GHz1.50GHz, 16GBRAM, where Python version 3.6 and Charm version 0.43. Since the public key generation scheme of this paper's scheme is the same as that of RW15, we only show the running efficiency of encryption and key generation, and decryption.

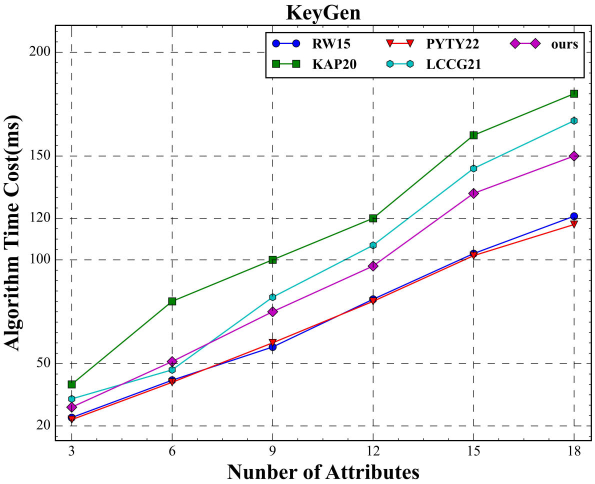

Figure 2. KeyGen

The first is the comparison of key generation time. According to Figure.2, it can be seen that our scheme is second only to RW15 and PYTY22 in terms of operation efficiency, and higher than other schemes. This is because, although all schemes are improved based on RW15, most of them have to introduce additional parameters in order to realize more complex functions. The scheme proposed in this paper, on the other hand, only adds a segment of key on top of the original scheme, which ensures that the operation efficiency is at an intermediate level and does not significantly slow down the execution efficiency of RW15.

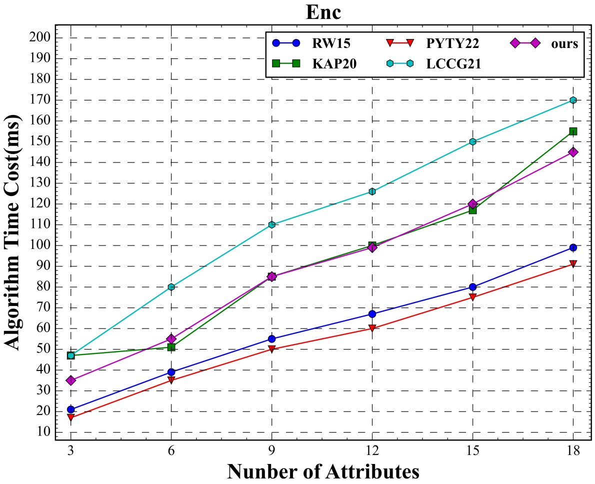

Figure 3. Encrypt

The second is the comparison of ciphertext generation time, similar to the key generation algorithm, the encryption algorithm of the scheme in this paper does not design additional ciphertext segments, and only performs one additional power and exponent operation. According to Figure.3, our scheme is in the middle position when the encryption algorithm runs, weaker than RW15 and PYTY22, and in a similar position with KAP20.

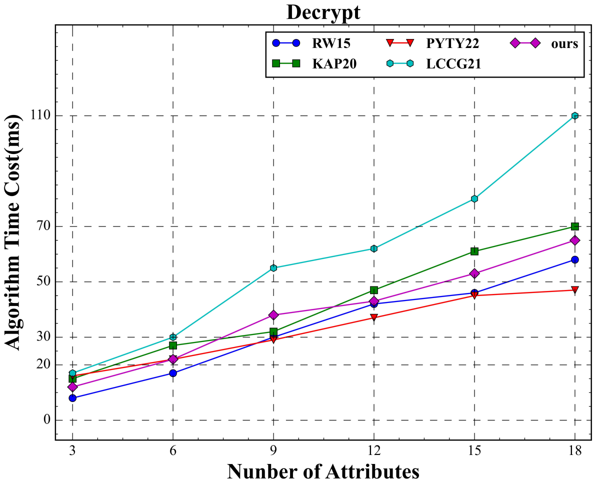

Figure 4. Decrypt

Finally, the cost time of the decryption algorithms, according to Figure.4, the decryption time consumed by the remaining algorithms converge and there is no significant efficiency gap except for LCCG21.

5. **Discuss.** In this paper, we propose a multi-authority attribute encryption algorithm for vehicular group networks, which solves the security problem caused by the variability of attribute permissions in group vehicular networks and achieves the revocation of group permissions. At the same time, the scheme avoids the risk of the distributed authority algorithm degrading into central authority due to the introduction of high-level attributes.

In the subsequent improvement work, the privilege revocation capability of the scheme can be enhanced to realize dynamic adaptive privilege update, which will further enhance the application potential of the scheme for group vehicle networking scenarios.

REFERENCES

[1] J. Cao, X. Di, J. Li, K. Yu, and L. Zhao, "Iovst: An anomaly detection method for iov based on spatiotemporal feature fusion," *Future Generation Computer Systems*, vol. 166, p. 107636, 2025.

[2] M. Adnan, M. Haider Syed, A. Anjum, and S. Rehman, "A framework for privacy-preserving in iov using federated learning with differential privacy," *IEEE Access*, vol. 13, pp. 13 507–13 521, 2025.

[3] X. Cai, J. Yang, T. Chang, Y. Fu, F. Richard Yu, N. Cheng, C. Li, and Y. Hui, "A reliable federated learning server rotation algorithm in iov," *IEEE Internet of Things Journal*, vol. 12, no. 11, pp. 15 501–15 513, 2025.

[4] C. Patel, A. Pasikhani, P. Gope, and J. Clark, "User-empowered secure privacy-preserving authentication scheme for digital twin," *Computers & Security*, vol. 140, p. 103793, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404824000944

[5] C.-M. Chen, Y. Hao, and T.-Y. Wu, "Discussion of "ultra super fast authentication protocol for electric vehicle charging using extended chaotic maps"," *IEEE Transactions on Industry Applications*, vol. 59, no. 2, pp. 2091–2092, 2023.

[6] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: Definitions and challenges," in *Theory of Cryptography: 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings 8*. Springer, 2011, pp. 253–273.

[7] A. O'Neill, "Definitional issues in functional encryption," Cryptology ePrint Archive, Paper 2010/556, 2010. [Online]. Available: https://eprint.iacr.org/2010/556

[8] B. Waters, “Efficient identity-based encryption without random oracles,” in *Advances in Cryptology–EUROCRYPT 2005: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005. Proceedings 24.* Springer, 2005, pp. 114–127.

[9] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in *Advances in Cryptology–EUROCRYPT 2005: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005. Proceedings 24.* Springer, 2005, pp. 457–473.

[10] D. Boneh and X. Boyen, “Efficient selective-id secure identity-based encryption without random oracles,” in *Advances in Cryptology-EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings 23.* Springer, 2004, pp. 223–238.

[11] D. Boneh and M. Franklin, “Identity-based encryption from the weil pairing,” *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[12] D. Yao, N. Fazio, Y. Dodis, and A. Lysyanskaya, “Id-based encryption for complex hierarchies with applications to forward security and broadcast encryption,” in *Proceedings of the 11th ACM Conference on Computer and Communications Security*, 2004, pp. 354–363.

[13] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,” in *2007 IEEE Symposium on Security and Privacy (SP'07).* IEEE, 2007, pp. 321–334.

[14] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, 2006, pp. 89–98.

[15] 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 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29.* Springer, 2010, pp. 62–91.

[16] A. Lewko and B. Waters, “Decentralizing attribute-based encryption,” in *Annual International Conference on the Theory and Applications of Cryptographic Techniques.* Springer, 2011, pp. 568–588.

[17] B. Waters, “Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization,” in *International Workshop on Public Key Cryptography.* Springer, 2011, pp. 53–70.

[18] V. Goyal, A. Jain, O. Pandey, and A. Sahai, “Bounded ciphertext policy attribute based encryption,” in *Automata, Languages and Programming: 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II 35.* Springer, 2008, pp. 579–591.

[19] J. Katz, A. Sahai, and B. Waters, “Predicate encryption supporting disjunctions, polynomial equations, and inner products,” in *Advances in Cryptology–EUROCRYPT 2008: 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings 27.* Springer, 2008, pp. 146–162.

[20] J. Lai, R. H. Deng, and Y. Li, “Fully secure cipertext-policy hiding cp-abe,” in *Information Security Practice and Experience: 7th International Conference, ISPEC 2011, Guangzhou, China, May 30–June 1, 2011. Proceedings 7.* Springer, 2011, pp. 24–39.

[21] T. Okamoto and K. Takashima, “Adaptively attribute-hiding (hierarchical) inner product encryption,” in *Advances in Cryptology–EUROCRYPT 2012: 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings 31.* Springer, 2012, pp. 591–608.

[22] J. Kim, W. Susilo, J. Baek, S. Nepal, and D. Liu, “Ciphertext-delegatable cp-abe for a dynamic credential: A modular approach,” in *Information Security and Privacy: 24th Australasian Conference, ACISP 2019, Christchurch, New Zealand, July 3–5, 2019, Proceedings 24.* Springer, 2019, pp. 3–20.

[23] Y. Ming, B. He, and C. Wang, “Efficient revocable multi-authority attribute-based encryption for cloud storage,” *IEEE Access*, vol. 9, pp. 42 593–42 603, 2021.

[24] K. Zhang, H. Li, J. Ma, and X. Liu, “Efficient large-universe multi-authority ciphertext-policy attribute-based encryption with white-box traceability,” *Science China Information Sciences*, vol. 61, pp. 1–13, 2018.

[25] S. Belguith, N. Kaaniche, M. Laurent, A. Jemai, and R. Attia, “Phoabe: Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted iot,” *Computer Networks*, vol. 133, pp. 141–156, 2018.

[26] Shruti, S. Rani, and W. Boulila, “Securing internet of things device data: An abe approach using fog computing and generative ai,” *Expert Systems*, vol. 42, no. 2, p. e13691, 2025.

[27] C. Li, J. He, C. Lei, C. Guo, and K. Zhou, “Achieving privacy-preserving cp-abe access control with multi-cloud,” in *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*. IEEE, 2018, pp. 801–808.

[28] S. J. De and S. Ruj, “Efficient decentralized attribute based access control for mobile clouds,” *IEEE Transactions on Cloud Computing*, vol. 8, no. 1, pp. 124–137, 2017.

[29] K. Sethi, A. Pradhan, and P. Bera, “Practical traceable multi-authority cp-abe with outsourcing decryption and access policy updation,” *Journal of Information Security and Applications*, vol. 51, p. 102435, 2020.

[30] J. Ling, J. Chen, J. Chen, and W. Gan, “Multiauthority attribute-based encryption with traceable and dynamic policy updating,” *Security and Communication Networks*, vol. 2021, no. 1, p. 6661450, 2021.

[31] Y. Rouselakis and B. Waters, “Efficient statically-secure large-universe multi-authority attribute-based encryption,” in *International Conference on Financial Cryptography and Data Security*. Springer, 2015, pp. 315–332.

[32] H. Yang, T. Feng, Y. Yan *et al.*, “Implementing efficient attribute encryption in iov under cloud environments,” *Computer Networks*, vol. 218, p. 109363, 2022.

[33] J. A. Akinyele, C. Garman, I. Miers, M. W. Pagano, M. Rushanan, M. Green, and A. D. Rubin, “Charm: a framework for rapidly prototyping cryptosystems,” *Journal of Cryptographic Engineering*, vol. 3, pp. 111–128, 2013.