Registration-Based Bilateral Fine-Grained Access Control in Vehicular Social Networks

Yang Ming, *Member, IEEE*, Hongyu Wang, Chenhao Wang, Hang Liu, Jie Feng, and Keqin Li, *Fellow, IEEE*

***Abstract*—Vehicular social networks (VSNs), as an innovative mobile communication system, significantly enhance the driving experience and improve urban traffic management efficiency. To address the privacy issues that arise from the use of public channels in VSNs, fine-grained access control (AC) should be ensured. Nevertheless, existing schemes still face some practical challenges in the aspects of data source identification, key escrow, and dynamic vehicle management. Therefore, this article proposes a registration-based bilateral fine-grained AC scheme (RBF-AC) in VSNs. Specifically, RBF-AC allows service providers (SPs) to select target vehicles and offer tailored services, while allowing vehicles to identify the most suitable SPs based on their needs, and all of which are realized in a fine-grained level. Meanwhile, SPs and vehicles are capable of locally generating their own private and public keys without relying on any fully trusted authority. RBF-AC also keeps high flexibility and enables the vehicles to join, leave and update their attributes in a dynamic manner. Additionally, outsourced verification and decryption are provided to minimize the computation cost for vehicles. We present formal security proofs to validate the security of RFB-AC. The performance evaluation illustrates the practical applicability of RBF-AC in VSNs.**

***Index Terms*—Fine-grained access control, matchmaking encryption (ME), registration-based cryptography, vehicular social networks (VSNs), without fully trusted authority.**

I. Introduction

IN THE context of rapidly evolving science and technology, as the fusion of Internet of Vehicles (IoV) and social networks, vehicular social networks (VSNs) are profoundly transforming modes of transportation and social interaction [1], [2], [3]. Through 5G and other communication technologies, service providers (SPs) can provide all kinds of services to vehicles by sharing traffic data and social information. This instant interaction not only optimizes traffic flow and public transportation efficiency, but also enriches users' daily experiences and improves their driving satisfaction [4], [5], [6]. Fig. 1 presents a specific architecture of VSNs, which mainly include SP, vehicles, roadside units (RSUs), and vehicle cloud server (VCS). SPs act as data senders by uploading service information to VCS. Meanwhile, vehicles download interested service data from the VCS through RSUs. As the processing center and coordinator, VCS handles large-scale data storage, performs data filtering, and conducts data preprocessing operations.

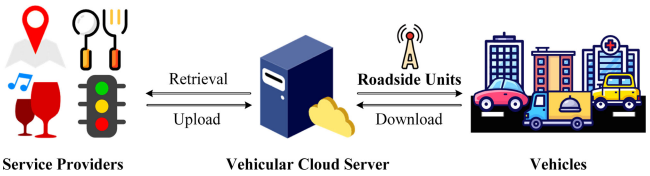

Fig. 1. Architecture of VSNs.

While there are many conveniences in VSNs, the data transmitted between SPs and vehicles may involve a large amount of private information, including driving routes, locations, and paid services, such as parking reservations. Once the data is leaked, it could pose a threat to users' safety and property. Furthermore, SP seeks to deliver specific information and services to multiple eligible vehicles with varying access privileges and attributes. Therefore, it is necessary to accomplish data privacy and fine-grained access control (AC).

Attribute-based encryption (ABE) [7], [8], [9], [10], [11] is a promising technology that can fulfill the aforementioned requirements. It enables SPs to define access policies tailored to requirements, ensuring that only vehicles with attributes meeting these policies can access the data. However, ABE is insufficient to meet the practical demands of VSNs. Specifically, the following issues and requirements should be considered.

1) *Bilateral Fine-Grained AC:* SP offers various services for vehicles, including traffic information, route planning, parking queries, and weather forecasts. However, the large amount of irrelevant content makes it difficult for vehicles to efficiently extract the information they actually need. In situations where a vehicle only requires traffic information for its current route, it is unnecessary to receive and process the information from multiple areas. As a result, vehicles expect to receive accurate and reliable data from SP. If a navigation system shows a clear road that is actually blocked, the

Received 25 February 2025; revised 18 May 2025; accepted 26 May 2025. Date of publication 29 May 2025; date of current version 25 July 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62472049 and Grant 62072054; in part by the Key Research and Development Program of Shaanxi Province under Grant 2024GX-YBXM-078 and Grant 2025SYS-SYSZD-080; and in part by the Applied Basic Research Foundation under Grant 2025A1515012741. *(Corresponding author: Chenhao Wang.)*

Yang Ming, Hongyu Wang, Chenhao Wang, and Hang Liu are with the School of Information Engineering, Chang'an University, Xi'an 710064, China (e-mail: yangming@chd.edu.cn; hywang@chd.edu.cn; chwang@chd.edu.cn; hangliu@chd.edu.cn).

Jie Feng is with the State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China (e-mail: jiefengcl@163.com).

Keqin Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Digital Object Identifier 10.1109/JIOT.2025.3574707

vehicle may mistakenly choose that route. These issues of irrelevant information and inaccuracy not only cause inconvenience for vehicles but may also lead to safety concerns. Therefore, vehicles must also enforce AC over SPs to select valid data. Namely, bilateral fine-grained AC between vehicles and SPs should be ensured.

2) *Without Fully Trusted Authority and Secure Channels:* On the one hand, it is unrealistic to establish a fully trusted authority (TA) in VSNs [12], [13]. Specifically, TA needs to generate secret keys for SPs and vehicles. As a result, the security of all data in VSNs will be severely threatened once TA is compromised. On the other hand, establishing secure channels for key distribution requires high costs and increases the complexity of the system. Therefore, fully TA and secure channels ought to be avoided.
3) *Dynamic Vehicle Management:* The joining, leaving, and attribute updates of vehicles in VSNs occur frequently as a result of their high mobility, posing significant challenges to network management and security. Specifically, when a new vehicle joins the network, it must be properly integrated into the system to ensure its access to network resources. Meanwhile, there are vehicles that leave the network due to disconnecting, disabling, and other reasons. The departing vehicles must be prohibited from accessing subsequent data even though their attributes fulfill the requirements of SPs. Additionally, the attributes of vehicles (such as appearance, usage, and ownership) may change due to transactions or modifications, which further increases the complexity of management. Consequently, the evolving changes in VSNs call for an efficient dynamic vehicle management mechanism.

Nevertheless, the existing schemes can only address some of the aforementioned issues and fail to provide a comprehensive solution. To solve issue 1), bilateral fine-grained AC schemes [14], [15], [16], [17], [18], [19], [20], [21], [22] were proposed inspired by ABE and matching encryption. These schemes not only encrypt data with predefined access policies but also incorporate data source identification and authenticity verification. Despite these advantages, [14], [15], [16], [17], [18], [19], [20], [21], [22] still rely on a TA and fail to achieve dynamic vehicle management, which leaves issues 2) and 3) unresolved.

Existing solutions address issue 2) by dividing the responsibilities of TA [23], [24] or removing the key distribution capability [25], [26]. Whereas, either bilateral AC or data authenticity is not satisfied in these approaches, causing issue 1) to remain unresolved. Although bilateral fine-grained AC schemes can be combined with [23], [24] or [25], [26] to simultaneously address both issues 1) and 2), the straightforward way faces challenges. On the one hand, [23], [24] introduce an additional authority, which must remain independent and noncolluding with the key distribution entity, posing practical difficulties in deployment. On the other hand, [25] is constructed based on the composite-order pairing groups, leading to additional security considerations. While [26] has limited expressiveness and only supports and-gate access policies.

In addressing issue 3), [27], [28] integrated attribute updating functionality into ABE to accommodate attribute changes, and the exit of users is supported in [29]. Additionally, [30] offers a more comprehensive solution by providing dynamic membership management, which supports user joining, exiting, and attribute updates. However, a common drawback of these schemes is the high computational cost, and they fail to address issues 1) and 2).

Therefore, the following question emerges: *"Can we design a bilateral fine-grained AC scheme in VSNs that supports dynamic vehicle management without fully TA and secure channels?"*

A. Contribution

This article aims to answer the above question positively by proposing a new solution named registration-based bilateral fine-grained AC (RBF-AC). The underling idea of RBF-AC comes from registered ABE (RABE) [25]. To be specific, SPs and vehicles generate the secret keys on their own and register the public keys and associated attributes with VCS. Here, VCS represents a key curator (KC) to aggregate all public keys of SPs and vehicles into a master public key. At the same time, KC is a transparent and semi-honest entity that maintains no secret and does not require secure channels to generate secret keys for users. Furthermore, the key contributions are as follows.

1) In RBF-AC, bilateral fine-grained AC between vehicles and SPs is guaranteed. Specifically, SPs are allowed to bind their own attributes and encrypt the data according to predefined access policies. In this way, SPs can precisely filter the target vehicles, ensuring that only vehicles whose attributes fulfill the requirements can retrieve service data. Meanwhile, vehicles are capable of defining policies to filter out irrelevant ciphertext, keeping only the valid data they need.
2) RBF-AC replaces the fully TA with a KC, which does not hold any secret value. Furthermore, SPs and vehicles can perform key generation locally, and KC is only responsible for key aggregation and public parameter management. Additionally, RBF-AC does not require secure channels, which significantly reduces deployment complexity and costs.
3) Efficient and dynamic vehicle management is provided in RBF-AC. When the vehicles join, leave, and attributes change, RBF-AC can update the corresponding parameters with the minimal computational cost, ensuring the system operates smoothly and continuously. Moreover, the large-scale computation tasks is offloaded to the cloud server, allowing vehicles to incur fixed decryption cost.
4) We have rigorously proven the confidentiality and authenticity of RBF-AC. Additionally, the evaluation of computation and storage cost demonstrates the superior performance of RBF-AC.

B. Organization

Section II and III present the related work and preliminaries, respectively. In Section IV, the system model, design

goals, syntax, and security model of RBF-AC are described. Section V presents the details of RBF-AC. In Section VI, the security proof of RBF-AC is elaborated. Section VII analyzes the performance of RBF-AC. Finally, this article is concluded in Section VIII.

II. Related Work

This section reviews the related work, including VSNs and the main cryptographic technologies, i.e., matchmaking encryption (ME) and registration-based encryption (RBE).

A. Vehicular Social Networks

For achieving the necessary security, efficiency, and functionality in AC for VSNs, Chen et al. [31] introduced a searchable encryption (SE) scheme for VSNs, which allows vehicles to perform ciphertext searches using keywords. However, only one-to-one AC is supported in [31]. To address this issue, a key-aggregate SE scheme was proposed by Sun et al. [32] to achieve secure one-to-many AC in VSNs. Nonetheless, [31], [32] only support single-keyword searches and do not meet the requirement for conjunctive keyword searches in VSNs. In consequence, Liu et al. [33] presented a data sharing system supporting conjunctive keyword searches, which significantly enhances the practicality of vehicle information retrieval. Additionally, a secure data sharing framework for VSNs was proposed in Fan et al. [34], which makes use of ABE to enable more data AC. However, the security of vehicle keys will be broken once TA is compromised. Motivated by this problem, Liu et al. [35] proposed an AC scheme that supports vehicle revocation without key escrow, ensuring that the key generation center remains unaware of the true keys associated with vehicles.

B. Matchmaking Encryption

Ateniese et al. [36] presented ME, which enables the sender to specify receivers through an access structure. At the same time, ME allows the receiver to verify if the ciphertext originates from an established sender. Furthermore, a generic framework for ME was given in [36] using function encryption and zero-knowledge proof techniques. Subsequently, identity-based ME (IBME) was constructed under the stochastic predicate machine model, and related schemes [37], [38], [39] were introduced to improve the security, efficiency, and functionality. Xu et al. [14] put forward attribute-based ME (ABME) to ensure bilateral fine-grained AC. Then, an improved scheme [15] was proposed by Sun et al.. based on [14], which effectively withstands forgery attacks and achieves bilateral fine-grained AC. Considering the problem of key leakage, Xu et al. [16] constructed an AC scheme, which realizes revocation of data recipients by sending the key update parameters of the current moment to legitimate users. Zhao et al. [17] developed a data source identification mechanism for vehicle platoons by combining ABE and inner-product encryption, which has a more efficient revocation mechanism compared to [16]. Additionally, the bilateral AC schemes initiated by Zhao et al. [18] and Hu et al. [19] not only support user revocation but also place significant emphasis on the protection of policy privacy. A new ME scheme using the anonymous credentials technique was suggested in Ma et al. [20], which employs blind signatures and zero-knowledge proofs to hide the sender's attributes for privacy protection, but the scheme does not guarantee attribute privacy when decryption fails. Bao et al. [21] posited a lightweight AC scheme that prevents adversaries from inferring sensitive information by concealing access policies, but it only supports and-gate policies. In Wu et al. [22], a bilateral AC scheme with expressive access policies was built. This scheme supports "AND/OR" operations in the access policy, which provides greater expressiveness than the and-gate, but it also incurs a higher computation cost.

TABLE I
Notations

Notations	Descriptions
U	Attribute universe
L	Number of users
S	Attribute set of user
S'	Updated attribute set of user
pk_i	Public key for user i
sk_i	Private key for user i
sp	The service providers
$\mathbb{S}$	Policy for controlling sp
$S \models \mathbb{S}$	S is consistent with $\mathbb{S}$
$S \not\models \mathbb{S}$	S is inconsistent with $\mathbb{S}$
v	The vehicles
$\mathbb{R}$	Policy for controlling v
tg	Interest tag
mpk	Master public key
aux	Auxiliary key

C. Registration-Based Encryption

To tackle the inherent issue of IBE, Garg et al. [40] presented RBE, where a KC is introduced to replace the traditional key generation center. Subsequently, [41] and [42] extended RBE to enhance anonymity and verifiability. However, the constructions of these schemes make nonblack-box usage of cryptographic primitives. [43] presented by Glaeser et al. utilizes a black-box model to increase computation efficiency while ensuring the preservation of both anonymity and verifiability. Compared to [40] and [43] eliminates the constraints on the order of user registration. Hohenberger et al. [25] introduced a RABE scheme in composite-order groups, achieving fine-grained AC. Francati et al. [26] extended [25] by constructing a registered functional encryption scheme that additionally supports policy hiding functionality.

III. Preliminaries

This section describes the relevant notations throughout this article (as described in Table I) and preliminary knowledge.

A. Bilinear Pairing

Given the cyclic groups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$, a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ fulfills the following characteristics for any $a, b \in \mathbb{Z}_p$, $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$.

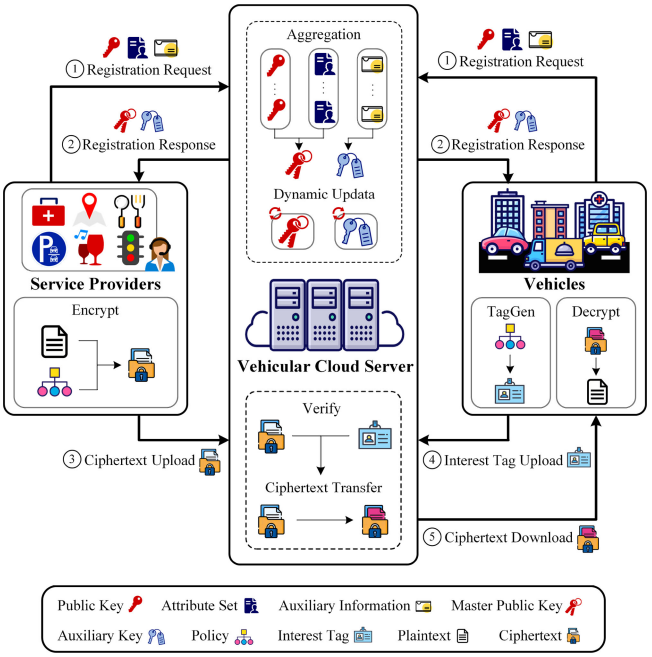

Fig. 2. System model.

1) *Bilinearity:* $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$ holds.
2) *Nondegeneracy:* $e(g_1, g_2) \neq 1$.
3) *Computability:* $e(g_1, g_2)$ can be calculated efficiently.

Our construction relies on Type II bilinear groups, where there is an publicly computable isomorphism $\varphi : \mathbb{G}_2 \to \mathbb{G}_1$.

B. Complexity Assumptions

1) *Decisional bilinear Diffie–Hellman (DBDH) assumption [38]:* Given the tuple $\{g_1^a, g_1^b, g_1^c, g_2^a, g_2^b, g_2^c, \}$ $\{e(g_1, g_2)^{abc}, e(g_1, g_2)^z\}$, where $a, b, c, z \in \mathbb{Z}_p$, $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$, distinguishing $e(g_1, g_2)^{abc}$ from $e(g_1, g_2)^z$ is hard for any probabilistic polynomial time (PPT) algorithm.
2) *Symmetric external Diffie–Hellman (SXDH) assumption [44]:* Given the tuple $\{g_2, g_1, g_1^a, g_1^b, g_1^{ab}, g_1^z\}$, where $a, b, z \in \mathbb{Z}_p$, $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$, distinguishing g_1^{ab} from g_1^z is difficult for any PPT algorithm.
3) *Computational Diffie–Hellman (CDH) assumption [22]:* Given the tuple $\{g_1^a, g_2^b\}$, where $a, b \in \mathbb{Z}_p$, $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$, the probability of calculating g_1^{ab} is negligible for any PPT algorithm.

IV. System Architecture

In this section, the system architecture of RBF-AC in VSNs is described, including system model, design goal, syntax, and security model.

A. System Model

Fig. 2 shows the system model of RBF-AC, which consists of vehicular cloud server (VCS), SPs, and vehicles.

1) *VCS:* VCS is a transparent and semi-honest entity and responsible for releasing public parameter strings. VCS receives the public keys and attribute sets from users (SPs and vehicles) ①, generates master public key and auxiliary key, and distributes them to SPs and vehicles ②. VCS is also responsible for updating the parameters when the attributes of users change or the vehicles exit VSNs. Additionally, since VCS possesses sufficient storage and computation resource, it filters ciphertext that meets the requirements of vehicles and conducts partial decryption for verified ciphertexts to reduce the computation cost of vehicles.
2) *SPs:* SPs generate the public/private key locally and upload the corresponding public keys and attribute sets to VCS. Meanwhile, SPs encrypt the plaintext data with predefined access policies and mater public key to control the vehicles and upload the ciphertext to VCS ③.
3) *Vehicles:* As the data receivers, vehicles should upload the necessary public keys, attribute sets, and auxiliary keys to VCS. Additionally, vehicles are able to upload the interest tags generated based on predefined access policies and mater public key to VCS ④, then download the corresponding ciphertext from VCS and perform decryption ⑤.

B. Design Goal

RBF-AC should satisfy the following goals:

1) *Security:* To prevent the leakage of privacy-sensitive information, data should be transmitted in ciphertext to VCS and can only be revealed in plaintext by authorized vehicles. Additionally, the authenticity of data must be ensured to prevent vehicles from being misled by erroneous information.
2) *Functionality:* RBF-AC should achieve fine-grained bilateral AC between SPs and vehicles, without fully TA and secure channels, and dynamic vehicle management, including joining, leaving, and attribute updates.
3) *Efficiency:* Considering the limited computational power of vehicles, RBF-AC should minimize the computation cost to ensure its practicality in VSNs.

C. Synatx

RBF-AC is composed of the following PPT algorithms.

1) $Setup(\lambda, U, L) \to crs$: Inputting the security parameter λ, the attribute universe U, and the number of users L, it generates the common reference string crs.
2) $KeyGen(i) \to (pk_i, sk_i)$: Inputting the index $i \in \{1, \ldots, L\}$, it generates the public key pk_i and the private key sk_i.
3) $Aggregate(\{(pk_i, S_i)\}_{i\in[1,L]}) \to (mpk, aux)$: Inputting the public keys and attribute sets (pk_i, S_i) for $i \in [1, L]$, it checks the validity of pk_i and generates the master public key mpk and the auxiliary key aux.
4) $Encrypt(mpk, aux, \mathbb{R}, m, sk_{sp}, S_{sp}) \to ct$: Inputting the master public key mpk, auxiliary key aux, access policy $\mathbb{R}$, plaintext m, the SP's private key sk_{sp}, and attribute set S_{sp}, it generates the ciphertext ct.
5) $TagGen(mpk, \mathbb{S}) \to tg$: Inputting the master public key mpk and access policy $\mathbb{S}$, it generates the interest tag tg.

6) *Verify*$(ct, tg) \rightarrow \{0, 1\}$: Inputting the ciphertext ct and interest tag tg, it outputs either 0 or 1.
7) *Transfer*$(ct) \rightarrow \overline{ct}$: Inputting the ciphertext ct, it generates the transferred ciphertext $\overline{ct}$.
8) *Decrypt*$(sk_v, \overline{ct}) \rightarrow m$: Inputting the vehicle's private key sk_v and the transferred ciphertext $\overline{ct}$, it generates the plaintext m.
9) *Update*$(mpk, aux, S, S') \rightarrow (mpk', aux')$: Inputting the master public key mpk, auxiliary key aux, attribute set S, and updated attribute set S' of the user, it generates the updated parameters (mpk', aux').
10) *Leave*$(mpk, aux, sk_v, S_v) \rightarrow (mpk', aux')$: Inputting the master public key mpk, auxiliary key aux, private key sk_v, and attribute set S_v of a vehicle that exits VSNs, it generates the updated master public key mpk' and the updated auxiliary key aux'.

D. Security Model

In RBF-AC, VCS is considered as a semi-honest entity, meaning it follows the protocol while also potentially being curious about the plaintext. SPs may be malicious, and they are capable of impersonating unauthorized users to generate false data; meanwhile, vehicles are regarded as untrustworthy, possibly colluding with others to decrypt unauthorized ciphertext. We require RBF-AC to ensure both data confidentiality and authenticity. Specifically, only vehicles that are authorized and registered within the system are allowed to access plaintext, and SPs cannot be impersonated.

The following game between the adversary $\mathcal{A}$ and the challenger $\mathcal{B}$ is introduced to define confidentiality.

Initialization: $\mathcal{A}$ chooses a challenging access policy $\mathbb{R}^*$ and sends $\mathbb{R}^*$ to $\mathcal{B}$.

Setup: $\mathcal{B}$ executes the algorithm *Setup* to generate *crs* and returns *crs* to $\mathcal{A}$.

Query Phase: The queries that $\mathcal{A}$ can perform are as follows.

1) *Key Generation Query:* Given an index $i \in [1, L]$, $\mathcal{B}$ executes the algorithm *KeyGen* to obtain (pk_i, sk_i) and returns pk_i to $\mathcal{A}$.
2) *Aggregation Query:* Given the tuples (i, S_i, pk_i) for $i \in [1, L]$, $\mathcal{B}$ executes the algorithm *Aggregate* to compute (*mpk*, *aux*) and returns them to $\mathcal{A}$.
3) *Corruption Query:* Given a index $i \in [1, L]$, $\mathcal{B}$ returns sk_i to $\mathcal{A}$.

Challenge: $\mathcal{A}$ chooses and sends two equal-length plaintexts (m_0^*, m_1^*) to $\mathcal{B}$. Then, a bit $b \in \{0, 1\}$ is selected by $\mathcal{B}$ to compute the challenge ciphertext ct^*.

Guess: $\mathcal{A}$ outputs a bit b', if $b = b'$ and sk_i is involved in the corruption query with $S_i \not\models \mathbb{R}^*$, $\mathcal{A}$ wins. $Adv^{\text{Conf}}_{\mathcal{A},\text{RBF-AC}}$ describes the advantage of $\mathcal{A}$ in winning the game.

Definition 1 (Confidentiality): RBF-AC satisfies confidentiality if the advantage $\text{Adv}^{\text{Conf}}_{\mathcal{A},\text{RBF-AC}}$ is negligible for the PPT adversary $\mathcal{A}$.

The following game between the adversary $\mathcal{A}$ and the challenger $\mathcal{B}$ is introduced to define authenticity.

Setup: $\mathcal{B}$ selects a index $i^* \in [1, L]$, executes the algorithm *Setup* to generate *crs*, and returns it to $\mathcal{A}$.

Query Phase: The queries that $\mathcal{A}$ can perform are as follows.

1) *H Query:* Given a plaintext m, $\mathcal{B}$ returns the result $H(m)$.
2) *Key Generation Query:* Given an index $i \in [1, L]$, $\mathcal{B}$ executes the algorithm *KeyGen* to obtain (pk_i, sk_i) and returns pk_i to $\mathcal{A}$.
3) *Aggregation Query:* Given the tuples (i, S_i, pk_i) for $i \in [1, L]$, $\mathcal{B}$ executes the algorithm *Aggregate* to compute (*mpk*, *aux*) and returns them to $\mathcal{A}$.
4) *Encryption Query:* Given a tuple $(i, S_i, \mathbb{R}, m)$, $\mathcal{B}$ executes the algorithm *Encrypt* to generate ct and returns it to $\mathcal{A}$ if $i \neq i^*$. Otherwise, $\mathcal{B}$ returns $\perp$.
5) *Corruption Query:* Given an index $i \in [1, L]$, $\mathcal{B}$ returns sk_i to $\mathcal{A}$ if $i \neq i^*$. Otherwise, $\mathcal{B}$ returns $\perp$.

Forgery: $\mathcal{A}$ forges ciphertext ct^* corresponding to (i, S_i^*, m^*). If $i = i^*$ and ct^* is a valid ciphertext, $\mathcal{A}$ wins. $Adv^{\text{Auth}}_{\mathcal{A},\text{RBF-AC}}$ describes the advantage of $\mathcal{A}$ in winning the game.

Definition 2 (Authenticity): RBF-AC satisfies authenticity if the advantage $Adv^{\text{Auth}}_{\mathcal{A},\text{RBF-AC}}$ is negligible for the PPT adversary $\mathcal{A}$.

V. Registration-Based Bilateral Fine-Grained Access Control Scheme

This section presents the details of the proposed registration-based bilateral fine-grained AC scheme (RBF-AC) in VSNs.

A. Workflow of RBF-AC

Fig. 3 illustrates the workflow of RBF-AC. At the beginning, VCS executes the algorithm *Setup* to generate the common reference string *crs*. Then, SPs and vehicles invoke the algorithm *KeyGen* locally to produce private keys sk_{sp} and sk_v, along with their public keys pk_{sp} and pk_v. Subsequently, the public keys together with attribute sets S_{sp} and S_v of SPs and vehicles are uploaded to VCS, which executes the algorithm *Aggregate* to validate pk_{sp} and pk_v. Once the public keys are valid, VCS generates the master public key mpk and auxiliary key aux for SPs and vehicles to complete the system's registration phase, which signifies that SPs and vehicles have joined the system.

Subsequently, SPs employ the algorithm *Encrypt* to generate the ciphertext ct, while vehicles invoke the algorithm *TagGen* to produce the interest tag tg that specifies the data they wish to access. VCS uses the algorithm *Verify* to match the interest tag with the ciphertext, filtering out ct that is relevant to the preferences of vehicles. To reduce the computation burden for resource-limited vehicles, the ciphertext ct is processed by VCS through the algorithm *Transfer* to generate the transferred ciphertext $\overline{ct}$, and VCS delivers $\overline{ct}$ to a target vehicle. In this way, the complex decryption process is outsourced to VCS. At the same time, the target vehicle can use the algorithm *Decrypt* to recover the plaintext m from $\overline{ct}$ with simple exponentiation operations.

Additionally, when vehicles need to leave the system or update the attributes, VCS executes the algorithms *Update* and *Leave* to generate the updated parameters. To be specific, the algorithms *Update* and *Leave*, respectively, generate the

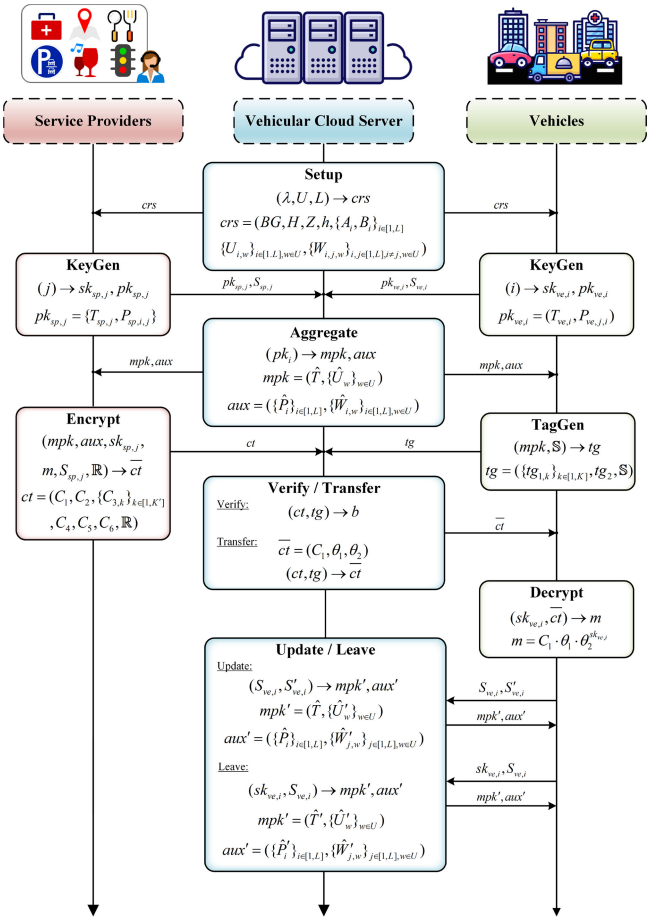

Fig. 3. Workflow of RBF-AC.

updated master public key mpk' and updated auxiliary key aux' to ensure the system can operate normally.

B. Concrete Construction of RBF-AC

In RBF-AC, SPs and vehicles have equal status and use the same key structure.

1) *Setup*: Given the security parameter λ, the attribute universe U, and the number of users L, VCS generates asymmetric bilinear groups $BG = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e)$, selects $\alpha, \beta \in \mathbb{Z}_p$, and computes $Z = e(g_1, g_2)^\alpha$, $h = g_1^\beta$. For $w \in U$ and $i, j \in [1, L]$ with $j \neq i$, VCS randomly selects $t_i, u_{i,w} \in \mathbb{Z}_p$, computes $A_i = g_2^{t_i}$, $B_i = g_2^\alpha A_i^\beta$, $U_{i,w} = g_1^{u_{i,w}}$, and $W_{i,j,w} = A_i^{u_{j,w}}$. VCS picks a hash function $H : \{0,1\}^* \to \mathbb{G}_2$ and outputs $crs = (BG, H, Z, h, \{A_i, B_i\}_{i\in[1,L]}, \{U_{i,w}\}_{i\in[1.L],w\in U}, \{W_{i,j,w}\}_{i,j\in[1,L],i\neq j,w\in U})$.
2) *KeyGen*: Any user (SP or vehicle) randomly selects $sk_i \in \mathbb{Z}_p$ and computes $pk_i = \{T_i = g_1^{sk_i}, P_{j,i} = A_j^{sk_i}\}$ for $j \in [1, L], j \neq i$.
3) *Aggregate*: Given the public key and attribute set $\{pk_i, S_i\}_{i\in[1,L]}$, VCS checks $e(g_1, P_{j,i}) = e(T_i, A_j)$ for $j \in [1, L], j \neq i$, computes

$$\widehat{T} = \prod_{i\in[1,L]} T_i, \quad \widehat{P}_i = \prod_{j\in[1,L],j\neq i} P_{i,j}$$

$$\widehat{U}_w = \prod_{i\in[1,L],w\notin S_i} U_{i,w}, \quad \widehat{W}_{i,w} = \prod_{j\in[1,L],j\neq i,w\notin S_j} W_{i,j,w}$$

and outputs $mpk = (\widehat{T}, \{\widehat{U}_w\}_{w\in U})$ and $aux = (\{\widehat{P}_i\}_{i\in[1,L]}, \{\widehat{W}_{i,w}\}_{i\in[1,L],w\in U})$.
4) *Encrypt*: Given the master public key mpk, auxiliary key aux, access policy $\mathbb{R} = (\mathbf{M} \in \mathbb{Z}_p^{K\times N}, \rho)$, private key $sk_{sp,j}$, and attribute set $S_{sp,j}$ of sp_j, where $j \in L$, $\mathbf{M}$ is a matrix, and the function ρ establishes a mapping between the rows of matrix $\mathbf{M}$ and the corresponding attributes. SP_j randomly selects $s, \tau \in \mathbb{Z}_p$, $h_1, h_2 \in \mathbb{G}_1$ such that $h = h_1 \cdot h_2$, generates a vector $\vec{x} = (x_1, x_2, \ldots, x_N)^T$ and defines $\vec{\lambda} = \mathbf{M} \cdot \vec{x} = (\lambda_1, \lambda_2, \ldots, \lambda_K)^T$, where $x_1 = s$ and $x_2, \ldots, x_N \in \mathbb{Z}_p$. SP_j then computes

$$C_1 = m \cdot e(g_1, g_2)^{\alpha s}, C_2 = g_1^s$$
$$C_{3,k} = h_2^{\lambda_k} \cdot \widehat{U}_{\rho(k)}^{-s}, C_4 = h_1^s \cdot \widehat{T}^{-s}, C_5 = g_1^\tau$$
$$C_6 = H(C_1 \| \cdots \| C_5)^\tau \cdot A_{sp,j}^{sk_{sp,j}}$$

and outputs $ct = (C_1, C_2, \{C_{3,k}\}_{k\in[1,K]}, C_4, C_5, C_6, \mathbb{R})$.
5) *TagGen*: Given the master public key mpk and access policy $\mathbb{S} = (\mathbf{N}, \pi)$, where $\mathbf{N} \in \mathbb{Z}_p^{K'\times N'}$, v_i samples $\hat{h}_1, \hat{h}_2 \in \mathbb{G}_1$ such that $h = \hat{h}_2 \cdot \hat{h}_2$, generates a vector $\vec{y} = (y_1, y_2, \ldots, y_{N'})^T$, and defines $\mu = \mathbf{N} \cdot \vec{y} = (\mu_1, \mu_2, \ldots, \mu_{K'})^T$, where $y_1 = 1$ and $y_2, \ldots, y_{N'} \in \mathbb{Z}_p$. v_i then computes

$$tg_{1,k} = \hat{h}_1^{\mu_k} \cdot \widehat{U}_{\pi(k)}^{-1}, tg_2 = \hat{h}_2 \cdot \widehat{T}^{-1}$$

and outputs $tg = (\{tg_{1,k}\}_{k\in[1,K']}, tg_2, \mathbb{S})$.
6) *Verify*: Given the ciphertext ct, interest tag tg, auxiliary key aux, and attribute set $S_{sp,j}$ of SP_j, VCS defines a set $\widehat{R}_1 = \{k \in [1, K']:\pi(k) \in S_{sp,j}\}$. If $S_{sp,j} \models \mathbb{S}$, then there exists a set of constants $\{w_k \in \mathbb{Z}_p\}_{k\in\widehat{R}_1}$ such that $\sum_{k\in\widehat{R}_1} w_k\mu_k = 1$. VCS then checks

$$e(A_{sp,j}, h) = \prod_{k\in\widehat{R}_1} (e(tg_{1,k}, A_{sp,j}) \cdot e(g_1, \widehat{W}_{sp,j,\rho(k)})^{w_k} \cdot e(tg_2, A_{sp,j}) \cdot e(g_1, C_6 \cdot \widehat{P}_{sp,j}) \cdot e(C_5, H(C_1 \| \cdots \| C_5))^{-1}$$

and outputs 1 if it holds. Otherwise, VCS outputs 0.
7) *Transfer*: Given the ciphertext ct, auxiliary key aux, and attribute set $S_{v,i}$ of v_i, VCS defines the set $\widehat{R}_2 = \{k|k \in [1, K], \rho(k) \in S_{v,i}\}$. If $S_{v,i} \models \mathbb{R}$, there exists a set of constants $\{w_k \in \mathbb{Z}_p\}_{k\in\widehat{R}_2}$ such that $\sum_{k\in\widehat{R}_2} w_k\lambda_k = s$. VCS then computes

$$\theta_1 = \prod_{k\in\widehat{R}_2} (e(C_{3,k}, A_{v,i}) \cdot e(C_2, \widehat{W}_{v,i,\rho(k)})^{w_k} \cdot e(C_4, A_{v,i}) \cdot e(C_2, \widehat{P}_{v,i}) \cdot e(C_2, B_{v,i})^{-1}$$
$$\theta_2 = e(C_2, A_{v,i})$$

and outputs $\overline{ct} = (C_1, \theta_1, \theta_2)$.
8) *Decrypt*: Given the private key $sk_{v,i}$ of v_i and transferred ciphertext $\overline{ct}$, v_i computes the plaintext $m = C_1 \cdot \theta_1 \cdot \theta_2^{sk_{v,i}}$.
9) *Update*: Given the attribute set S_i, updated attribute set S_i', master public and auxiliary keys (mpk, aux), VCS defines the sets $\Gamma_1 = S_i - S_i'$ and $\Gamma_2 = S_i' - S_i$, where Γ_1

represents the set of attributes that are in set S_i but not in set S_i', and Γ_2 represents the set of attributes that are in set S_i' but not in set S_i. For all $w \in \Gamma_1,\ j \in [1, L], j \neq i$, VCS computes

$$\widehat{U}_w' = \widehat{U}_w \cdot U_{i,w}, \quad \widehat{W}_{j,w}' = \widehat{W}_{j,w} \cdot W_{j,i,w}$$

For all $w \in \Gamma_2,\ j \in [1, L], j \neq i$, VCS computes

$$\widehat{U}_w' = \widehat{U}_w \cdot U_{i,w}^{-1}, \quad \widehat{W}_{j,w}' = \widehat{W}_{j,w} \cdot W_{j,i,w}^{-1}.$$

VCS outputs $(\{\widehat{U}_w'\}_{w\in U},\ \{\widehat{W}_{j,w}'\}_{j\in[1,L],w\in U})$.

10) *Leave*: Given the public key pk_i, attribute set S_i, master public key *mpk*, and auxiliary key *aux*, VCS computes

$$\widehat{T}' = \widehat{T} \cdot pk_i^{-1},\ \widehat{U}_w' = \widehat{U}_w \cdot U_{i,w}^{-1},$$
$$\widehat{P}_j' = \widehat{P}_j \cdot P_{j,i}^{-1},\quad \widehat{W}_{j,w}' = \widehat{W}_{j,w} \cdot W_{j,i,w}^{-1}.$$

for $j \in [1, L], j \neq i, w \notin S_i$ and outputs $(\{\widehat{U}_w'\}_{w\in U}, \{\widehat{W}_{j,w}'\}_{j\in[1,L],w\in U},\ \widehat{T}', \{\widehat{P}_j'\}_{j\in[1,L]})$.

Correctness of algorithm *Verify*:

$$\begin{aligned}
\Theta_1 &= e\big(tg_2, A_{sp,j}\big) \cdot e\big(g_1, C_6 \cdot \widehat{P}_{sp,j}\big) \cdot e(C_5, H(CT_{1-5})^{-1} \\
&= e\Big(\hat{h}_2 \cdot \widehat{T}^{-1}, A_{sp,j}\Big) \cdot e\Big(g_1, \widehat{A}_{sp,j}^{sk_{sp,j}} \cdot \widehat{P}_{sp,j}\Big) \\
&\quad \cdot e\big(g_1, H(C_1 \| \cdots \| C_5)^{\tau}\big) \cdot e\big(g_1^{\tau}, H(C_1 \| \cdots \| C_5)\big)^{-1} \\
&= e\Big(\hat{h}_2 \cdot \widehat{T}^{-1}, A_{sp,j}\Big) \cdot e\Big(g_1, A_{sp,j}^{sk_{sp,j}} \cdot \widehat{P}_{sp,j}\Big) \\
&= e\Big(\hat{h}_2, A_{sp,j}\Big) \cdot e\Big(\widehat{T}^{-1}, A_{sp,j}\Big) \cdot e\Big(g_1, A_{sp,j}^{sk_{sp,j}} \cdot \widehat{P}_{sp,j}\Big) \\
&= e\Big(\hat{h}_2, A_{sp,j}\Big) \cdot e\left(\left(\prod_{i\in[1,L]} g_1^{sk_i}\right)^{-1}, g_2^{t_j}\right) \\
&\quad \cdot e\left(g_1, g_2^{t_j sk_{sp,j}} \cdot \prod_{i\in[1,L],i\neq j} g_2^{t_j sk_i}\right) \\
&= e\Big(\hat{h}_2, A_{sp,j}\Big) \\
\Theta_2 &= \prod_{k\in\widehat{R}_1} \big(e(tg_{1,k}, A_{sp,j}) \cdot e(g_1, \widehat{W}_{sp,j,\pi(k)})\big)^{w_k} \\
&= \prod_{k\in\widehat{R}_1} \Big(e(\hat{h}_1^{\mu_k}, A_{sp,j}) \cdot e(\widehat{U}_{\pi(k)}^{-1}, A_{sp,j}) \cdot e(g_1, \widehat{W}_{sp,j,\pi(k)})\Big)^{w_k} \\
&= \prod_{k\in\widehat{R}_1} \left(e\Big(\hat{h}_1^{\mu_k}, A_{sp,j}\Big) \cdot e\left(\prod_{l\in[1,L],\pi(k)\notin S_l} g_1^{-u_{l,\pi(k)}}, g_1^{t_j}\right)\right. \\
&\quad \left. \cdot e\left(g_1, \prod_{l\in[1,L]\setminus\{j\},\pi(k)\notin S_l} g_1^{t_j\cdot u_{l,\pi(k)}}\right)\right)^{w_k} \\
&= e\Big(\hat{h}_1, A_{sp,j}\Big) \\
\Theta_1 \cdot \Theta_2 &= \prod_{k\in\widehat{R}_1} \big(e(tg_{1,k}, A_{sp,j}) \cdot e(g_1, \widehat{W}_{sp,j,\rho(k)})\big)^{w_k} \\
&\quad \cdot e\big(tg_2, A_{sp,j}\big) \cdot e\big(g_1, C_6 \cdot \widehat{P}_{sp,j}\big) \cdot e(C_5, H(CT_{1-5}))^{-1} \\
&= e\big(h, A_{sp,j}\big).
\end{aligned}$$

Correctness of algorithm *Decrypt*:

$$\begin{aligned}
\Theta_3 &= \prod_{k\in\widehat{R}_2} \big(e(C_{3,k}, A_{v,i}) \cdot e(C_2, \widehat{W}_{v,i,\rho(k)})\big)^{w_k} \cdot e\big(C_4, A_{v,i}\big) \\
&\quad \cdot e\big(C_2, \widehat{P}_{v,i}\big) \cdot e\big(C_2, B_{v,i}\big)^{-1} \cdot e\big(C_2, A_{v,i}\big)^{sk_{v,i}} \\
&= \prod_{k\in\widehat{R}_2} (e(h_2^{\lambda_k} \cdot \widehat{U}_{\rho(k)}^{-s}, A_{v,i} \cdot e(g_1^s, \widehat{W}_{v,i,\rho(k)}))^{w_k} \\
&\quad \cdot e(h_1^s \cdot \widehat{T}^{-s}, g_2^{t_i}) \cdot e(C_2, \widehat{P}_{v,i}) \cdot e(g_1^s, g_2^{\alpha})^{-1} \\
&\quad \cdot e(g_1^s, g_2^{\beta t_i})^{-1} \cdot e(g_1^s, g_2^{t_i})^{sk_{v,i}} \\
&= e(h_2^s, A_{v,i}) \cdot \prod_{k\in\widehat{R}_2} (e(\widehat{U}_{\rho(k)}^{-s}, A_{v,i}) \cdot e(g_1^s, \widehat{W}_{v,i,\rho(k)}))^{w_k} \\
&\quad \cdot e(h_1^s, g_2^{t_i}) \cdot e(\widehat{T}^{-s}, g_2^{t_i}) \cdot e(g_1^s, \widehat{P}_{v,i} \cdot g_2^{t_i \cdot sk_{v,i}}) \\
&\quad \cdot e(g_1^s, g_2^{\alpha})^{-1} \cdot e(h, g_2^{s\cdot t_i})^{-1} \\
&= e(h^s, A_{v,i}) \cdot e(h, g_2^{s\cdot t_i})^{-1} \cdot e(g_1^s, g_2^{\alpha})^{-1} \\
&\quad \cdot \prod_{k\in\widehat{R}_2} \left(e\left(\prod_{l\in[1,L]:\pi(k)\notin S_l} g_1^{-s\cdot u_{l,\pi(k)}}, g_2^{t_i}\right)\right. \\
&\quad \left. \cdot e\left(g_1^s, \prod_{l\in[1,L]\setminus\{i\}:\pi(k)\notin S_l} g_1^{t_i\cdot u_{l,\pi(k)}}\right)\right)^{w_k} \\
&= e(g_1^s, g_2^{\alpha})^{-1} \\
C_1 \cdot \Theta_3 &= m \cdot e(g_1, g_2)^{\alpha s} \cdot e(g_1^s, g_2^{\alpha})^{-1} = m.
\end{aligned}$$

C. Dynamicity of RBF-AC

In VSNs, vehicles are capable of dynamically joining and leaving the system and updating attributes due to their mobility. First, the algorithm *Aggregate* supports vehicle joining, but it is limited to one-time operations and does not support the dynamic addition of vehicles. Fortunately, RBF-AC can naturally leverage the generic compiler proposed in [25] to achieve dynamic addition, thereby integrating subsequent vehicles into the network. Second, when a vehicle leaves the system, VCS can run the algorithm *Leave* to adjust the current master public key *mpk* and auxiliary key *aux*. Third, when the attributes change, VCS executes the algorithm *Update* to adjust $\widehat{T}$ and $\widehat{U}_w$ associated with the attributes. This is achieved by defining Γ_1 and Γ_2, which describe the difference of attributes change.

VI. Security Analysis

This section proves the confidentiality and authenticity of RBF-AC.

Theorem 1 (Confidentiality): If the DBDH and SXDH assumptions hold, RBF-AC satisfies confidentiality.

Proof: The following games are utilized to prove the confidentiality of RBF-AC.

Game 0: It is the initial confidentiality game as described in Section IV.

Initialization: $\mathcal{A}$ chooses $\mathbb{R}^* = (\mathbf{M}^* \in \mathbb{Z}_p^{n^*\times l^*}, \rho^*)$ and sends $\mathbb{R}^*$ to $\mathcal{B}$.

Setup: $\mathcal{B}$ samples $a, b, \zeta, t_i', \alpha', u_{i,w} \in \mathbb{Z}_p$ and creates two empty lists $L_{key} : \{i, pk_i, sk_i\}$ and $L_{cor} : \{i, pk_i\}$. For $i, j \in [1, L]$, $i \neq j$, and $w \in U$, $\mathcal{B}$ computes *crs* as follows:

$$Z = e(g_1, g_2)^{ab}, h = g_1^{\zeta a}, A_i = g_2^{t_i' - b/\zeta}$$
$$B_i = g_2^{ab+\alpha'} \cdot A_i^{\zeta a} = g_2^{ab+\alpha'} \cdot g_2^{\zeta a(t_i' - b/\zeta)}$$
$$U_{i,w} = g_1^{u_{i,w}}, W_{i,j,w} = A_i^{u_{j,w}} = g_2^{(t_i' - b/\zeta)\cdot u_{j,w}}.$$

Finally, $\mathcal{B}$ returns $(Z, h, \{A_i, B_i\}_{i\in[1,L]}, \{U_{i,w}\}_{i\in[1.L],w\in U}, \{W_{i,j,w}\}_{i,j\in[1,L],i\neq j,w\in U})$ to $\mathcal{A}$.

Query Phase: $\mathcal{A}$ performs queries as follows:

1) *Key Generation Query:* Given the user index $i \in [1, L]$, $\mathcal{B}$ samples $sk_i \in \mathbb{Z}_p$, computes $pk_i = (g_1^{sk_i}, g_2^{(t_i'-b/\xi)\cdot sk_i})$, returns pk_i to $\mathcal{A}$, and adds (i, pk_i, sk_i) to L_{key}.
2) *Aggregation Query:* Given the tuples $(i, S_i, \widehat{pk_i})$ for $i \in [1, L]$, $\mathcal{B}$ checks if $\{i, \widehat{pk_i}\} \in L_{key}$. If so, $\mathcal{B}$ runs the algorithm *Aggregate* to compute $\{\widehat{T}, \{\widehat{U}_w\}_{w\in U}, \{\widehat{P}_i\}_{i\in[1,L]}, \{\widehat{W}_{i,w}\}_{i\in[1,L],w\in U}\}$. Otherwise, $\mathcal{B}$ proceeds as follows.
   a) If $\widehat{pk_i}$ is valid, $\mathcal{B}$ adds i to L_{cor} and runs the algorithm *Aggregate* to compute $\{\widehat{T}, \{\widehat{U}_w\}_{w\in U}, \{\widehat{P}_i\}_{i\in[1,L]}, \{\widehat{W}_{i,w}\}_{i\in[1,L],w\in U}\}$.
   b) Otherwise, $\mathcal{B}$ returns $\perp$.

   Finally, $\mathcal{B}$ returns $mpk = (\widehat{T}, \{\widehat{U}_w\}_{w\in U})$ and $aux = (\{\widehat{P}_i\}_{i\in[1,L]}, \{\widehat{W}_{i,w}\}_{i\in[1,L],w\in U})$ to $\mathcal{A}$.
3) *Corruption Query:* Given the user index $i \in [1, L]$, $\mathcal{B}$ looks up L_{key} to obtain (i, pk_i, sk_i), returns sk_i to $\mathcal{A}$, and adds (i, pk_i) to L_{cor}.

Challenge: $\mathcal{A}$ outputs (m_0^*, m_1^*), $\mathcal{B}$ chooses $b \in \{0, 1\}$ and computes ct^* as follows:

$$C_1 = m_b^* \cdot e(g_1, g_2)^{abs}, C_2 = g_1^s$$
$$C_{3,k} = h_2^{\lambda_k^*} \cdot \widehat{U}_{\rho(k)}^{-s} = g_1^{\chi\cdot\lambda_k^*} \cdot \left(\prod_{i\in[1,L],\ w\notin S_i} U_{i,w}\right)^{-s}$$
$$C_4 = h_1^s \cdot \widehat{T}^{-s} = g_1^{(\zeta a-\chi)\cdot s} \cdot \left(\prod_{i\in[1,L]} T_i\right)^{-s}, C_5 = g_1^{\tau}$$
$$C_6 = H(C_1\| \cdots \|C_5)^{\tau} \cdot g_2^{t_i\cdot sk_i}.$$

where $\lambda_k^* = M_k^* \cdot \vec{x}$, M_k^* is the k-th row of M^*, $\vec{x} = (s, x_2, \ldots, x_{l^*})^T \in \mathbb{Z}_p^{l^*}$, $\xi \in \mathbb{Z}_p^*$, and $h_2 = g_1^{\xi}$, $h_1 = g_1^{\zeta a-\xi}$ such that $h_1 \cdot h_2 = h$. Finally, $\mathcal{B}$ returns $ct^* = (C_1, C_2, \{C_{3,k}\}_{k\in[1,n^*]}, C_4, C_5, C_6)$.

Guess: $\mathcal{A}$ outputs b' and wins the game if $b = b' \wedge i \in L_{cor}, S_i \not\models \mathbb{R}^*$, i.e., *Game 0*$(\mathcal{A}) = 1$.

Game 1: It is the same as *Game 0* except that $\mathcal{B}$ chooses $z_1 \in \mathbb{Z}_p$ as well as computes and returns ct^* as follows:

$$C_1 = m_b^* \cdot e(g_1, g_2)^{abs}, C_2 = g_1^s,$$
$$C_{3,k} = g_1^{\chi\cdot\lambda_k^*} \cdot \left(\prod_{i\in[1,L],\ w\notin S_i} U_{i,w}\right)^{-s}$$
$$C_4 = g_1^{z_1 a-\chi s} \cdot \left(\prod_{i\in[1,L]} T_i\right)^{-s}, C_5 = g_1^{\tau}$$
$$C_6 = H(C_1\| \cdots \|C_5)^{\tau} \cdot g_2^{t_i\cdot sk_i}.$$

Guess: $\mathcal{A}$ outputs b' and wins the game if $b = b' \wedge i \in L_{cor}, S_i \not\models \mathbb{R}^*$, i.e., *Game 1*$(\mathcal{A}) = 1$. ∎

Lemma 2: If the SXDH assumption holds, for the PPT adversary $\mathcal{A}$, $|\Pr[\textit{Game 0}(\mathcal{A}) = 1] - \Pr[\textit{Game 1}(\mathcal{A}) = 1]|$ is negligible.

Proof: $g_1^{\zeta\cdot s\cdot a} \approx g_1^{z_1\cdot a}$ can be concluded according to the SXDH assumption. More precisely, $g_1^{(\zeta a-\chi)\cdot s} \cdot (\prod_{i\in[1,L]} \widehat{T}_i)^{-s}$ and $g_1^{z_1 a-\chi s} \cdot (\prod_{i\in[1,L]} \widehat{T}_i)^{-s}$ are computationally indistinguishable. In conclusion, the ciphertexts of *Game 0* and *Game 1* are indistinguishable.

Game 2: It is the same as *Game 1* except that $\mathcal{B}$ chooses $z_2 \in \mathbb{Z}_p$ as well as computes and returns ct^* as follows:

$$C_1 = m_b^* \cdot e(g_1, g_2)^{z_2}, C_2 = g_1^s$$
$$C_{3,k} = g_1^{\chi\cdot\lambda_k^*} \cdot \left(\prod_{i\in[1,L],\ w\notin S_i} U_{i,w}\right)^{-s}$$
$$C_4 = g_1^{z_1 a-\chi s} \cdot \left(\prod_{i\in[1,L]} T_i\right)^{-s}, C_5 = g_1^{\tau}$$
$$C_6 = H(C_1\| \cdots \|C_5)^{\tau} \cdot g_2^{t_i\cdot sk_i}.$$

Guess: $\mathcal{A}$ outputs b' and wins the game if $b = b' \wedge i \in L_{cor}, S_i \not\models \mathbb{R}^*$, i.e., *Game 2*$(\mathcal{A}) = 1$. ∎

Lemma 3: If the DBDH assumption holds, for the PPT adversary $\mathcal{A}$, $|\Pr[\textit{Game 1}(\mathcal{A}) = 1] - \Pr[\textit{Game 2}(\mathcal{A}) = 1]|$ is negligible.

Proof: It can be obtained that $e(g_1, g_2)^{abs} \approx e(g_1, g_2)^{z_2}$ according to the DBDH assumption. Specifically, $m_b^* \cdot e(g_1, g_2)^{abs}$ and $m_b^* \cdot e(g_1, g_2)^{z_2}$ are computationally indistinguishable. In conclusion, the ciphertext of *Game 1* and *Game 2* are indistinguishable.

Based on the Lemmas 2 and 3, we can conclude that the ciphertexts of *Game 0* and *Game 2* are computationally indistinguishable. Concretely, $m_b^* \cdot e(g_1, g_2)^{abs}$ and $m_b^* \cdot e(g_1, g_2)^{z_2}$ cannot be distinguished from each other. In *Game 2*, the ciphertext $m_b^* \cdot e(g_1, g_2)^{z_2}$ is also indistinguishable from a random group element. Therefore, the probability that $\mathcal{A}$ wins is $1/2$.

Theorem 4 (Authenticity): If the CDH assumption holds, RBF-AC satisfies authenticity.

Proof: If the adversary $\mathcal{A}$ can break the authenticity of RBF-AC, then we can build an algorithm $\mathcal{B}$ to solve the CDH problem. Namely, given the tuple $\{g_1^a, g_2^b\}$, $\mathcal{B}$ is aimed at computing g_1^{ab}.

Setup: $\mathcal{B}$ chooses a index $i^* \in [1, L]$, samples α, β, $t_i, u_{i,w} \in \mathbb{Z}_p$ for $i \in [1, L], w \in U$, and creates two empty lists $L_{key} = \{i, pk_i, sk_i\}$ and $L_H = \{m_i, y_i, H(m_i)\}$. Then, $\mathcal{B}$ computes crs for $i, j \in [1, L], i \neq j, w \in U$ as follows:

$$Z = e(g_1, g_2)^{\alpha} = e(g_1, g_2)^{\alpha}, \ h = g_1^{\beta}$$
$$A_i = \begin{cases} g_2^{t_i}, & i \neq i^* \\ g_2^{b}, & i = i^* \end{cases}, \quad B_i = \begin{cases} g_2^{\alpha}\cdot g_2^{\beta t_i}, & i \neq i^* \\ g_2^{\alpha}\cdot g_2^{\beta b}, & i = i^* \end{cases}$$
$$U_{i,w} = g_1^{u_{i,w}}, \quad W_{i,j,w} = \begin{cases} g_2^{u_{j,w}t_i}, & i \neq i^* \\ g_2^{u_{j,w}b}, & i = i^* \end{cases}.$$

Finally, $\mathcal{B}$ returns $(Z, h, \{A_i, B_i\}_{i\in[1,L]}, \{U_{i,w}\}_{i\in[1.L],w\in U}, \{W_{i,j,w}\}_{i,j\in[1,L],i\neq j,w\in U})$ to $\mathcal{A}$.

Query Phase: $\mathcal{A}$ performs queries as follows:

1) *H Query:* Given m_i, $\mathcal{B}$ returns $H(m_i)$ to $\mathcal{A}$ if $(m_i, y_i, H(m_i)) \in L_H$. Otherwise, $\mathcal{B}$ randomly samples $y_i \in \mathbb{Z}_p$, sets $H(m_i) = g_2^{y_i}$, returns $H(m_i)$ to $\mathcal{A}$, and adds $(m_i, y_i, H(m_i))$ to L_H.

2) *Key Generation Query:* Given the user index $i \in [1, L]$, $\mathcal{B}$ proceeds as follows.
   a) If $i \neq i^*$, $\mathcal{B}$ chooses $sk_i \in \mathbb{Z}_p$, computes $pk_i = g^{sk_i}$, returns pk_i to $\mathcal{A}$, and adds (i, pk_i, sk_i) to L_{key}.
   b) If $i = i^*$, $\mathcal{B}$ sets $sk_{i^*} = a$, $pk_{i^*} = g_1^{sk_{i^*}} = g_1^a$, returns pk_i to $\mathcal{A}$, and adds $(i, pk_{i^*}, \emptyset)$ to L_{key}.
3) *Aggregation Query:* Given the tuples $(i, S_i, \widehat{pk_i})$ for $i \in [1, L]$, $\mathcal{B}$ checks if $\{i, \widehat{pk_i}\} \in L_{key}$. If so, $\mathcal{B}$ runs the algorithm *Aggregate* to compute $\{\widehat{T}, \{\widehat{U}_w\}_{w\in U}, \{\widehat{P}_i\}_{i\in[1,L]}, \{\widehat{W}_{i,w}\}_{i\in[1,L],w\in U}\}$. Otherwise, $\mathcal{B}$ proceeds as follows.
   a) If $\widehat{pk_i}$ is valid, $\mathcal{B}$ adds i to L_{cor} and run the algorithm *Aggregate* to compute $\{\widehat{T}, \{\widehat{U}_w\}_{w\in U}, \{\widehat{P}_i\}_{i\in[1,L]}, \{\widehat{W}_{i,w}\}_{i\in[1,L],w\in U}\}$.
   b) Otherwise, $\mathcal{B}$ returns $\perp$.

   Finally $\mathcal{B}$ returns $mpk = (\widehat{T}, \{\widehat{U}_w\}_{w\in U})$ and $aux = (\{\widehat{P}_i\}_{i\in[1,L]}, \{\widehat{W}_{i,w}\}_{i\in[1,L],w\in U})$ to $\mathcal{A}$.
4) *Encryption Query:* Given a tuple $(i, S_i, \mathbb{R})$, $\mathcal{B}$ generates a vector $\vec{x} = (s, x_2, x_3, \ldots, x_l^*)^T \in \mathbb{Z}_p^{l^*}$, and computes $\lambda_k = F_k \cdot \vec{x}$, where F_k is the kth row of the matrix F. $\mathcal{B}$ randomly selects $\tau \in \mathbb{Z}_p$, $h_1, h_2 \in \mathbb{G}_1$ such that $h_1 \cdot h_2 = h$ and computes

$$C_1 = m \cdot e(g_1, g_2)^{\alpha s}, C_2 = g_1^s$$
$$C_{3,k} = h_2^{\lambda_k} \cdot \widehat{U}_{\rho(k)}^{-s} = h_2^{\lambda_k} \cdot \left(\prod_{i\in[1,L],\ w\notin S_i} U_{i,w} \right)^{-s}$$
$$C_4 = h_1^s \cdot \widehat{T}^{-s} = h_1^s \cdot \left(\prod_{i\in[1,L]} g_1^{sk_i} \right)^{-s}, C_5 = g_1^{\Omega}$$
$$C_6 = H(C_1 \| \cdots \| C_5)^{\Omega} \cdot A_i^{sk_i} = g_2^{y_i \Omega} \cdot g_2^{t_i \cdot sk_i}.$$

   Finally, $\mathcal{B}$ returns $ct = (C_1, C_2, \{C_{3,k}\}_{k\in[1,n^*]}, C_4, C_5, C_5, C_6)$ to $\mathcal{A}$.
5) *Corruption Query:* Given the user index $i \in [1, L]$, $\mathcal{B}$ looks up L_{key} to obtain (i, pk_i, sk_i) and returns sk_i to $\mathcal{A}$ if $i \neq i^*$. Otherwise, $\mathcal{B}$ returns $\perp$.

Forgery: $\mathcal{A}$ outputs ciphertext ct under (i, m^*). If $i \neq i^*$, $\mathcal{B}$ terminates. Otherwise, we have $H(C_1^* \| \cdots \| C_5^*) = g_2^{y_{i^*}}$.

According to the algorithm *Encrypt*, $C_5^* = g_1^{\tau'}$ and $C_6^* = H(C_1^* \| \cdots \| C_5^*)^{\tau'} \cdot A_{i^*}^{sk_{i^*}} = g_2^{\tau' y_{i^*}} g_2^{ab}$ can be concluded. Therefore, $\mathcal{B}$ can compute $\varphi(C_6^*) \cdot (C_5^*)^{-y_{i^*}} = \varphi(A_{i^*}^{sk_{i^*}}) = g_1^{ab}$, which contradicts the CDH assumption. ■

VII. Performance Evaluation

This section compares RBF-AC with the related schemes XNL+ [14], SYT+ [15], XNH+ [16], and WMZ+ [22] in terms of functionality, computation cost, and storage cost.

A. Functionality Comparison

Table II presents the functionality comparison between RBF-AC and [14], [15], [16], [22]. It is obvious that our RBF-AC can achieve more functions compared to [14], [15], [16], and [22]. Specifically, only our RBF-AC can achieve without fully TA, which not only reduces the cost of system establishment but also avoids the privacy leakage caused by the TA compromises. Nevertheless, there exist tradeoffs between security and flexibility in RBF-AC, i.e., it needs to predefine the number of users. Additionally, RBF-AC supports dynamic exit and attribute updates. In this way, the vehicles that leave the system no longer have access to the data even though their attributes satisfy the policies of SPs. Meanwhile, when the attributes of vehicles, such as appearance and ownership change, RBF-AC can update the vehicle attributes to adjust new access rights.

TABLE II
Functionality Comparison

Schemes	F1	F2	F3	F4	F5	F6	F7
XNL+ [14]	●	○	●	○	○	○	●
SYT+ [15]	●	○	●	○	○	○	●
XNH+ [16]	●	○	●	●	○	●	●
WMZ+ [22]	●	○	●	○	○	○	●
RBF-AC	●	●	●	●	●	●	○

F1: Bilateral fine-grained access control; F2: Without fully trusted authority; F3: Dynamic join; F4: Dynamic exit; F5: Attribute updates; F6: Outsourced decryption; F7: Unbounded user number; ● : Satisfied; ○ : Unsatisfied.

TABLE III
Notion and Description of Cryptographic Operations

Notations	Descriptions	Time (ms)
T_p	Bilinear pairing operation	5.6464
T_H	Map-to-point hash operation	23.4527
T_a	Addition operation in $\mathbb{G}$	0.0099
T_e	Exponentiation operation in $\mathbb{G}$	4.6956
$T_{e_{\mathbb{G}}}$	Exponentiation operation in $\mathbb{G}_T$	0.4951

B. Performance Comparison

The experiments were conducted on a 64-bit Ubuntu operating system using a personal computer equipped with an i5-10400 CPU running at 2.90 GHz and 4 GB of RAM. Under 128-bit level, Type-F pairing over a curve $E: y^2 = x^3 + x$ is selected.

Table III describes the running time of cryptographic operations. Furthermore, $|\mathbb{Z}_p| = 256$ bits, $|\mathbb{G}| = 3072$ bits, $|\mathbb{G}_T| = 3072$ bits denote the size of an element in $\mathbb{Z}_p$, $\mathbb{G}$, and $\mathbb{G}_T$. Table IV compares RBF-AC with XNL+ [14], SYT+ [15], XNH+ [16], and WMZ+ [22] for computation and storage cost.

According to Table IV, the computation and storage cost of all the schemes are mostly closely related to the size of the attribute set $|S|$. In terms of computation cost, XNL+, SYT+, and WMZ+ require a map-to-point hash operation for each attribute in the encryption algorithm, while it is not involved in XNH+ and RBF-AC. Therefore, as the size of attribute set $|S|$ increases, the encryption cost of RBF-AC grows more slowly compared to XNL+, SYT+, XNH+, and WMZ+. For the verification algorithm, RBF-AC requires fewer exponentiation operations in $\mathbb{G}$ compared to others since the simplicity of the encryption key. For the decryption algorithm, XNH+ and RBF-AC have fixed

TABLE IV
COMPARISON OF COMPUTATION AND STORAGE COST

Schemes	Computation Cost			Storage Cost		
	Encryption	Verification	Decryption	Encryption Key	Decryption Key	Ciphertext
XNL+ [14]	$(2\|S\|+3)T_e+(2\|S\|+1)T_a+(2\|S\|+1)T_H$ $=56.3164\|S\|+37.5494$	$(2\|S\|+1)T_p+\|S\|T_{e_\mathbb{G}}+(\|S\|+1)T_a+(\|S\|+1)T_H$ $=35.2505\|S\|+29.109$	$2\|S\|T_p+\|S\|T_{e_\mathbb{G}}+(\|S\|+1)T_a$ $=11.7978\|S\|+0.0099$	$(\|S\|+1)\|\mathbb{G}\|$ $=3072(\|S\|+1)$	$2\|S\|\|\mathbb{G}\|$ $=6144\|S\|$	$(3\|S\|+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$ $=9216\|S\|+9216$
SYT+ [15]	$(3\|S\|+2)T_e+(2\|S\|+1)T_a+(2\|S\|+1)T_H$ $=61.012\|S\|+32.8538$	$(2\|S\|+1)T_p+\|S\|T_{e_\mathbb{G}}+(\|S\|+1)T_a+(\|S\|+1)T_H$ $=35.2505\|S\|+29.109$	$2\|S\|T_p+\|S\|T_{e_\mathbb{G}}+(\|S\|+1)T_a$ $=11.7978\|S\|+0.0099$	$2\|S\|\|\mathbb{G}\|$ $=6144\|S\|$	$2\|S\|\|\mathbb{G}\|$ $=6144\|S\|$	$(3\|S\|+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$ $=9216\|S\|+9216$
XNH+ [16]	$(8\|S\|+8)T_e+(4\|S\|+6)T_a+T_H$ $=37.6044\|S\|+61.0769$	$(3\|S\|+2)T_p+2\|S\|T_e+\|S\|T_{e_\mathbb{G}}+3\|S\|T_a+T_H$ $=26.8552\|S\|+34.7455$	$T_{e_\mathbb{G}}+T_a$ $=0.5050$	$(2\|S\|+2)\|\mathbb{G}\|$ $=6144\|S\|+6144$	$\|\mathbb{Z}_p\|$ $=256$	$(5\|S\|+4)\|\mathbb{G}\|+\|\mathbb{G}_T\|$ $=15360\|S\|+15360$
WMZ+ [22]	$(4\|S\|+2)T_e+(\|S\|+1)T_a+(2\|S\|)T_H$ $=65.6977\|S\|+9.4011$	$4\|S\|T_p+4\|S\|T_{e_\mathbb{G}}+(2\|S\|)T_a$ $=24.5858\|S\|$	$2\|S\|T_p+\|S\|T_{e_\mathbb{G}}+(\|S\|+1)T_a$ $=11.7978\|S\|+0.0099$	$(\|S\|+1)\|\mathbb{G}\|+\|\mathbb{Z}_p\|$ $=3072\|S\|+3328$	$(2\|S\|+1)\|\mathbb{G}\|$ $=6144\|S\|+3072$	$(2\|S\|+3)\|\mathbb{G}\|+\|\mathbb{G}_T\|$ $=6144\|S\|+12288$
RBF-AC	$(2\|S\|+7)T_e+(\|S\|+3)T_a+T_H$ $=9.4011\|S\|+56.3506$	$(2\|S\|+4)T_p+(\|S\|+1)T_{e_\mathbb{G}}+(\|S\|+3)T_a+T_H$ $=11.7978\|S\|+46.5631$	$T_{e_\mathbb{G}}+2T_a$ $=0.5149$	$\|\mathbb{Z}_p\|$ $=256$	$\|\mathbb{Z}_p\|$ $=256$	$(\|S\|+4)\|\mathbb{G}\|+\|\mathbb{G}_T\|$ $=3072\|S\|+15360$

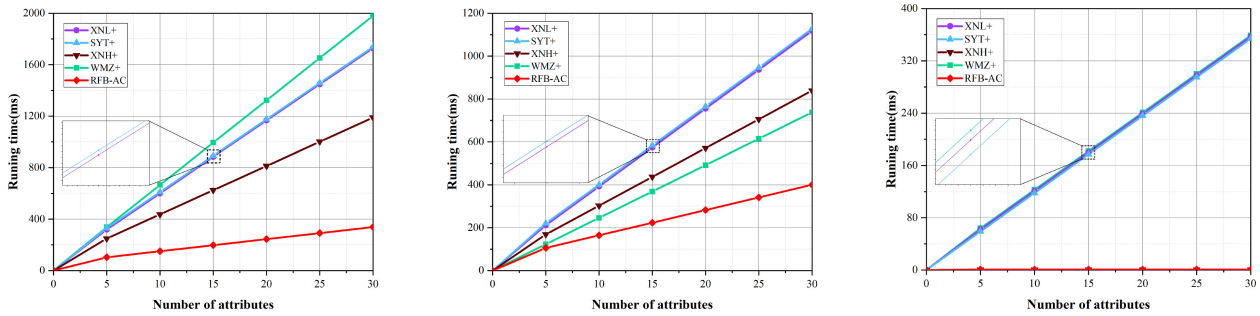

Fig. 4. Computation cost. (a) Computation cost of encryption. (b) Computation cost of verification. (c) Computation cost of decryption

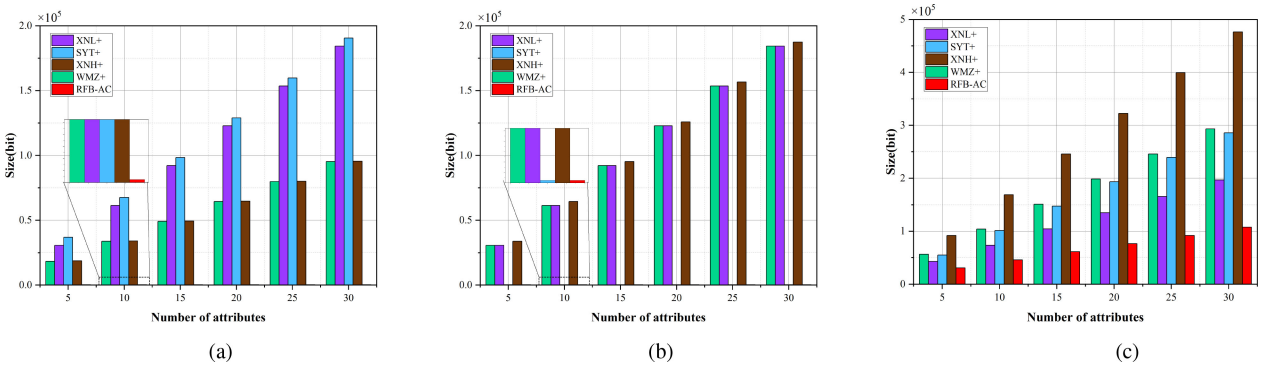

Fig. 5. Storage cost. (a) Storage cost of encryption key. (b) Storage cost of decryption key. (c) Storage cost of ciphertext.

computation cost, which is much smaller than XNL+, SYT+, and WMZ+. The main reason is that the outsourcing computation technology is adopted in XNH+ and RBF-AC, where a large amount of decryption cost is outsourced to the cloud server. Regarding storage cost, RBF-AC achieves a fixed-size encryption key because all the parameters are included in the common reference string. In contrast, the size of the encryption key in XNL+, SYT+, XNH+, and WMZ+ depends on $|S|$. Additionally, XNH+ and RBF-AC have a fixed-size decryption key because the matching of attributes and policies is performed by the VCS. Compared with XNL+, SYT+, XNH+, and WMZ+, RBF-AC has the lowest ciphertext storage cost.

Fig. 4 presents the running time of the encryption, verification and decryption algorithms. Specifically, Fig. 4(a) indicates that the running time of the encryption algorithm in all the schemes is proportional to the number of attributes. Nevertheless, RBF-AC has the slowest growth rate. When the number of attributes is 30, the time required for SPs to generate the ciphertext in XNL+, SYT+, XNH+, WMZ+, and RBF-AC is 1738.1, 1762.3, 1189.2, 1980.3, and 338.4 ms, respectively. RBF-AC saves 80.4%, 80.8%, 71.5%, and 82.9% compared to XNL+, SYT+, XNH+, and WMZ+. The running time of the verification algorithm in all the schemes is illustrated in Fig. 4(b). As the number of attributes increases, RBF-AC achieves the best efficiency. When there are 30

attributes, the time required for VCS to verify the ciphertext in XNL+, SYT+, XNH+, WMZ+, and RBF-AC is 1117.9, 1132.5, 839.5, 737.6, and 400.5 ms, separately. Compared with XNL+, SYT+, XNH+, and WMZ+, RBF-AC reduces 64.2%, 64.6%, 52.3%, and 45.7%. As illustrated in Fig. 4(c), the running time of the decryption algorithm in XNL+, SYT+, and WMZ+ grows as the number of attributes increases, while the outsourcing computing is adopted in XNH+ and RBF-AC. Therefore, the time required for vehicles to decrypt the ciphertext remains constant.

Fig. 5 illustrates the storage cost of the encryption key, decryption key, and ciphertext. Concretely, as shown in Fig. 5(a), the storage cost of encryption key in RBF-AC is independent with the size of attribute set $|S|$. In contrast, the storage cost of encryption key in XNL+, SYT+, XNH+, and WMZ+ is proportional to $|S|$. Therefore, RBF-AC maintains lower storage cost even though there exists a large attribute set. The storage cost of decryption key is depicted in Fig. 5(b), where the cost of XNL+, XNH+, and WMZ+ is related to $|S|$. On the contrary, SYT+ and RBF-AC have a fixed-size decryption key. Fig. 5(c) shows the ciphertext storage cost of XNL+, SYT+, XNH+, WMZ+, and RBF-AC. It is obvious that RBF-AC has the lowest cost. When $|S| = 30$, the ciphertext storage cost of XNL+, SYT+, XNH+, WMZ+, and RBF-AC is 196608, 285696, 476160, 293376, and 107520 bits, respectively. RBF-AC reduces the ciphertext storage cost by 45.3%, 62.4%, 77.4%, and 63.4% compared to XNL+, SYT+, XNH+, and WMZ+.

VIII. Conclusion

This article presents a RBF-AC in VSNs. On the one hand, RBF-AC achieves bilateral fine-grained AC. Namely, SP can accurately choose vehicles and offer customized services, while vehicles can precisely identify the required data from diverse information sources based on their needs. On the other hand, RBF-AC allows SPs and vehicles to generate the keys locally and does not require any fully TA and secure channels. In addition, efficient dynamic vehicle management is fulfilled in RBF-AC. Whether vehicles join, leave VSNs and attributes change, RBF-AC is capable of updating the corresponding parameters with minimal computational cost. Finally, we provide formal security proofs and performance evaluation to demonstrate that RFB-AC is highly suitable for VSNs and offers superior practicality and potentiality compared to existing schemes.

Although RBF-AC can achieve nice security features, there are other security challenges in VSNs to be solved, such as physical vehicle capture attacks and attribute privacy leakage. In future work, we will conduct more thorough investigations for the security challenges in VSNs. Furthermore, the corresponding solutions will be proposed, including forward-secure data transmission scheme and attribute-hiding data sharing scheme.

References

[1] X. Lin, R. Lu, C. Zhang, H. Zhu, P.-H. Ho, and X. Shen, "Security in vehicular ad hoc networks," *IEEE Commun. Mag.*, vol. 46, no. 4, pp. 88–95, Apr. 2008.

[2] A. M. Vegni and V. Loscrí, "A survey on vehicular social networks," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2397–2419, 4th Quart., 2015.

[3] H. Liu, Y. Ming, C. Wang, and Y. Zhao, "Flexible selective data sharing with fine-grained erasure in VANETs," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 9582–9597, 2024.

[4] M. Vinodhini and S. Rajkumar, "Performance analysis of vehicle-to-everything communication using Internet of LoRa computing for intelligent transportation system," *Intell. Decis. Technol.*, vol. 17, no. 2, pp. 577–594, 2023.

[5] X. Wang et al., "Privacy-preserving content dissemination for vehicular social networks: Challenges and solutions," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1314–1345, 2nd Quart., 2019.

[6] C. Wang, Y. Ming, H. Liu, J. Feng, M. Yang, and Y. Xiang, "Blockchain-assisted privacy-preserving and Synchronized key agreement for VDTNs," *IEEE Trans. Dependable Secure Comput.*, early access, Jan. 21, 2025, doi: 10.1109/TDSC.2025.3532287.

[7] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. 24th Adv. Cryptol. EUROCRYPT*, 2005, pp. 457–473.

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

[9] N. Chen, J. Li, Y. Zhang, and Y. Guo, "Efficient CP-ABE scheme with shared decryption in cloud storage," *IEEE Trans. Comput.*, vol. 71, no. 1, pp. 175–184, Jan. 2022.

[10] S. Chen, J. Li, Y. Zhang, and J. Han, "Efficient revocable attribute-based encryption with verifiable data integrity," *IEEE Internet Things J.*, vol. 11, no. 6, pp. 10441–10451, Mar. 2024.

[11] J. Li, E. Zhang, J. Han, Y. Zhang, and J. Shen, "PH-MG-ABE: A flexible policy-hidden multigroup attribute-based encryption scheme for secure cloud storage," *IEEE Internet Things J.*, vol. 12, no. 2, pp. 2146–2157, Jan. 2025.

[12] R. Zhang, J. Li, Y. Lu, J. Han, and Y. Zhang, "Key escrow-free attribute based encryption with user revocation," *Inf. Sci.*, vol. 600, pp. 59–72, Jul. 2022.

[13] C. Wang, Y. Ming, H. Liu, and Y. Deng, "Dual fine-grained authentication without trusted authority for data collection in TDT systems," *IEEE Trans. Mobile Comput.*, early access, Feb. 6, 2025, doi: 10.1109/TMC.2025.3539281.

[14] S. Xu et al., "Match in my way: Fine-grained bilateral access control for secure cloud-fog computing," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 2, pp. 1064–1077, Mar./Apr. 2022.

[15] J. Sun, Y. Yuan, M. Tang, X. Cheng, X. Nie, and M. U. Aftab, "Privacy-preserving bilateral fine-grained access control for cloud-enabled industrial IoT healthcare," *IEEE Trans. Ind. Informat.*, vol. 18, no. 9, pp. 6483–6493, Sep. 2022.

[16] S. Xu, J. Ning, X. Huang, J. Zhou, and R. H. Deng, "Server-aided bilateral access control for secure data sharing with dynamic user groups," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4746–4761, 2021.

[17] Y. Zhao et al., "Secure source identification scheme for revocable instruction sharing in vehicle platoon," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 5, pp. 3671–3684, May 2024.

[18] M. Zhao, C. Zhang, T. Wu, J. Ni, X. Liu, and L. Zhu, "Revocable and privacy-preserving bilateral access control for cloud data sharing," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 5389–5404, 2024.

[19] X. Hu, L. Wang, L. Gu, and Y. Ning, "A bilateral access control data sharing scheme for Internet of Vehicles," *IEEE Internet Things J.*, vol. 11, no. 22, pp. 36748–36762, Nov. 2024.

[20] J. Ma, S. Xu, J. Ning, X. Huang, and R. H. Deng, "Catch me if you can: A secure bilateral access control system with anonymous credentials," *IEEE Trans. Services Comput.*, vol. 16, no. 6, pp. 4444–4455, Nov./Dec. 2023.

[21] Y. Bao et al., "Lightweight and bilateral controllable data sharing for secure autonomous vehicles platooning service," *IEEE Trans. Veh. Technol.*, vol. 72, no. 11, pp. 13969–13984, Nov. 2023.

[22] T. Wu, X. Ma, C. Zhang, X. Liu, G. Yang, and L. Zhu, "Toward fine-grained task allocation with bilateral access control for intelligent transportation systems," *IEEE Internet Things J.*, vol. 11, no. 8, pp. 14814–14828, Apr. 2024.

[23] J. Hur, "Improving security and efficiency in attribute-based data sharing," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 10, pp. 2271–2282, Oct. 2013.

[24] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proc. ACM Conf. Comput. Commun. Security*, 2009, pp. 121–130.

[25] S. Hohenberger, G. Lu, B. Waters, and D. J. Wu, "Registered attribute-based encryption," in *Proc. Adv. Cryptol. EUROCRYPT*, 2023, pp. 511–542.
[26] D. Francati, D. Friolo, M. Maitra, G. Malavolta, A. Rahimi, and D. Venturi, "Registered (inner-product) functional encryption," in *Proc. 29th Adv. Cryptol. ASIACRYPT*, 2023, pp. 98–133.
[27] P. Zhang, Z. Chen, J. K. Liu, K. Liang, and H. Liu, "An efficient access control scheme with outsourcing capability and attribute update for fog computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 753–762, Jan. 2018.
[28] P. Zhang, Z. Chen, K. Liang, S. Wang, and T. Wang, "A cloud-based access control scheme with user revocation and attribute update," in *Proc. 21st Australas. Conf. Inf. Security Privacy*, 2016, pp. 525–540.
[29] C. Wang, Y. Ming, H. Liu, Y. Deng, Y. Zhao, and S. Zhang, "Security-enhanced data transmission with fine-grained and flexible revocation for DTWNs," *IEEE Trans. Inf. Forensics Security*, vol. 20, pp. 1237–1250, 2025.
[30] C.-I. Fan, V. S.-M. Huang, and H.-M. Ruan, "Arbitrary-state attribute-based encryption with dynamic membership," *IEEE Trans. Comput.*, vol. 63, no. 8, pp. 1951–1961, Aug. 2014.
[31] B. Chen, L. Wu, H. Wang, L. Zhou, and D. He, "A blockchain-based searchable public-key encryption with forward and backward privacy for cloud-assisted vehicular social networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5813–5825, Jun. 2020.
[32] J. Sun, H. Xiong, S. Zhang, X. Liu, J. Yuan, and R. H. Deng, "A secure flexible and tampering-resistant data sharing system for vehicular social networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12938–12950, Nov. 2020.
[33] H. Liu, Y. Ming, C. Wang, Y. Zhao, S. Zhang, and R. Lu, "Server-assisted data sharing system supporting conjunctive keyword search for vehicular social networks," *IEEE Trans. Services Comput.*, vol. 17, no. 6, pp. 4281–4295, Nov./Dec. 2024.
[34] K. Fan et al., "A secure and verifiable data sharing scheme based on blockchain in vehicular social networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5826–5835, Jun. 2020.
[35] T. Liu, Z. Li, Y. Ji, and J. Chang, "Efficient key-escrow-free and vehicle-revocable data sharing protocol for vehicular ad hoc network," *IEEE Internet Things J.*, vol. 11, no. 7, pp. 11540–11553, Apr. 2024.
[36] G. Ateniese, D. Francati, D. Nuñez, and D. Venturi, "Match me if you can: Matchmaking encryption and its applications," in *Proc. 39th Adv. Cryptol. CRYPTO*, 2019, pp. 701–731.
[37] J. Chen, Y. Li, J. Wen, and J. Weng, "Identity-based matchmaking encryption from standard assumptions," in *Proc. 28th Adv. Cryptol. ASIACRYPT*, 2022, pp. 394–422.
[38] J. Sun, G. Xu, T. Zhang, X. Yang, M. Alazab, and R. H. Deng, "Privacy-aware and security-enhanced efficient matchmaking encryption," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 4345–4360, 2023.
[39] Y. Cao, J. Wei, X. Huang, X. Chen, and Y. Xiang, "Deniable identity-based matchmaking encryption for anonymous messaging," *IEEE Trans. Dependable Secure Comput.*, vol. 22, no. 3, pp. 2197–2210, May/Jun. 2025, doi: 10.1109/TDSC.2024.3479236.
[40] S. Garg, M. Hajiabadi, M. Mahmoody, and A. Rahimi, "Registration-based encryption: Removing private-key generator from IBE," in *Proc. 16th Int. Conf. Theory Cryptogr.*, 2018, pp. 689–718.
[41] S. Garg, M. Hajiabadi, M. Mahmoody, A. Rahimi, and S. Sekar, "Registration-based encryption from standard assumptions," in *Proc. 22nd IACR Int. Conf. Pract. Theory Public-Key Cryptogr.*, 2019, pp. 63–93.
[42] R. Goyal and S. Vusirikala, "Verifiable registration-based encryption," in *Proc. 40th Adv. Cryptol. CRYPTO*, 2020, pp. 621–651.
[43] N. Glaeser, D. Kolonelos, G. Malavolta, and A. Rahimi, "Efficient registration-based encryption," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2023, pp. 1065–1079.
[44] J. Chen, H. W. Lim, S. Ling, H. Wang, and H. Wee, "Shorter IBE and signatures via asymmetric pairings," in *Proc. 5th Int. Conf. Pairing-Based Cryptogr.*, 2013, pp. 122–140.