Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3513063, 14 pages
https://doi.org/10.1155/2022/3513063

Research Article

Multiauthority Attribute-Based Keyword Search over Cloud-Edge-End Collaboration in IoV

Yan Zhen,[1,2,3,4] Yilan Chui,[1,2,3] Puning Zhang,[1,2,3] and Huan Liu[1,2,3]

[1]*School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China*
[2]*Advanced Network and Intelligent Interconnection Technology Key Laboratory of Chongqing Education Commission of China, China*
[3]*Chongqing Key Laboratory of Ubiquitous Sensing and Networking, Chongqing, China*
[4]*State Grid Information and Communication Industry Group Limited, Beijing, China*

Correspondence should be addressed to Puning Zhang; zhangpn@cqupt.edu.cn

Received 31 December 2021; Accepted 19 April 2022; Published 31 May 2022

Academic Editor: Changqing Luo

The development of cloud computing and edge computing makes it possible to store and share Internet of Vehicles (IoV) data on a large scale, which greatly contributes to traffic intelligence, but outsourced data confidentiality and user privacy cannot be guaranteed. The Ciphertext Policy Attribute-Based Encryption (CP-ABE) scheme can achieve both fine-grained access control and secure data sharing. However, existing CP-ABE schemes owns high computational complexity, and the adopted single attribute authority mode is burdensome to resource-limited IoV. Thus, this paper proposes a Multiauthority Attribute-Based Keyword Search over Cloud-Edge-End Collaboration (CEABKS-MA) system, leveraging the benefits of edge and cloud resources and effectively combining with the multiauthority structure to minimize the computation and storage pressure on resource-limited parties in the system. In addition, fine-grained keyword search with support for attribute update and lightweight encryption/decryption is extended. Finally, this paper demonstrates the security and efficiency of the CEABKS-MA system through rigorous security analysis and simulation experiments.

1. Introduction

IoV collects and transmits vehicle data to the network through in-vehicle sensing devices, which largely facilitates intelligent transportation system [1–3]. Currently, with the development of IoV technology and the increase in the number of vehicles, users' demands for intelligent access to vehicle information are increasing. Since IoV data usually includes users' private information (e.g., vehicle location), privacy protection becomes a key factor affecting IoV user search experience. However, in-vehicle networks exposed to unprotected environments are more vulnerable to security threats in data storage, transmission, and sharing [4, 5] and have limited computation and storage capacity to effectively support secure sharing of IoV data.

Outsourcing massive amounts of data to the cloud and edge can effectively alleviate the resource-limited issue of IoV devices and facilitate data sharing, but the cloud and edge are usually perceived as "honest and curious" [6, 7], meaning they will honor agreements honestly but may have unauthorized access to some sensitive data. Hence, the searchable encryption (SE) [8] is proposed to support outsourced data encryption and enable keyword search in the ciphertext domain, where special encryption algorithms are used by data owners and search users for encryption of plaintext data, indexes, and queries to perform accurate or near-accurate keyword matching operations on the ciphertext.

Facing the massive amount of IoV data, fine-grained access control to the data in secure retrieval is of great importance to users. The CP-ABE scheme [9, 10] embeds the access policy into the ciphertext; the user can decrypt the ciphertext only when the user attributes satisfy the policy; thus, the data owner can control the access to the data, which is ideal for dynamic IoV environments. However,

most existing CP-ABE schemes are mainly based on cloud computing architecture [11, 12] and are designed for single attribute authority scenarios [13–15] and have high computational complexity. Among them, centralized cloud computing imposes a heavy computation and storage burden on cloud servers, and remote cloud-oriented data transmissions cause high communication cost and large latency [16–18]. As user registration and key generation of single attribute authority are resource-intensive and time-consuming, which may lead to the failure of single-point attribute authority and serious consequences such as key and user privacy leakage, ultimately affecting the availability of search system. Besides, multiple types of sensing devices are deployed on vehicles to collect corresponding vehicle attribute data, while users may only need vehicle data for a certain attribute; traditional keyword search will return ciphertext for all attributes of the vehicle, which will bring additional computation consumption and communication overhead for users.

Compared with centralized cloud computing, cloud-edge-end collaborative architecture can effectively reduce search latency and the computational load on the cloud by fusing the advantages of the edge being closer to users and the cloud having abundant resources [19–21]. Figure 1 illustrates the IoV cloud-edge-end collaborative architecture. Therefore, this paper designs a multiauthority attribute keyword search (CEABKS-MA) system based on cloud-edge-end collaboration, which can effectively reduce the computation and storage burden of resource-constrained parties in the system and achieve efficient and secure fine-grained keyword retrieval for IoV data. The main contributions of this paper are as follows:

(i) Fine-grained keyword search with support for attribute update and lightweight encryption is extended in access control. Vehicles carrying multiple sensors are abstracted into one or more attributes to enable retrieval of specified vehicle attributes. The attribute update function effectively prevents malicious users from stealing who revoke attributes, and the online/offline encryption method further reduces the computation burden on users

(ii) A cloud-edge-end collaborative search method is proposed, where users can achieve real-time and historical search by sending a trapdoor to the nearest edge. Besides, the edge provides a ciphertext pre-decryption service, and the search user can obtain plaintext data by performing a simple calculation

(iii) A multiauthority structure is designed to implement distributed key management, which decentralizes the expensive and time-consuming key generation and distribution tasks of the central authority to each attribute authority, which can better adapt to the spatial characteristics of vehicles and distributed IoV topology

The main content of this paper is as follows. Section 2 discusses related work. Section 3 introduces preparatory knowledge. Section 4 presents the system model, formal definition, and security model. Section 5 describes in detail the construction of the CEABKS-MA system. Section 6 analyzes the security and performance of the CEABKS-MA system. Section 7 concludes this paper.

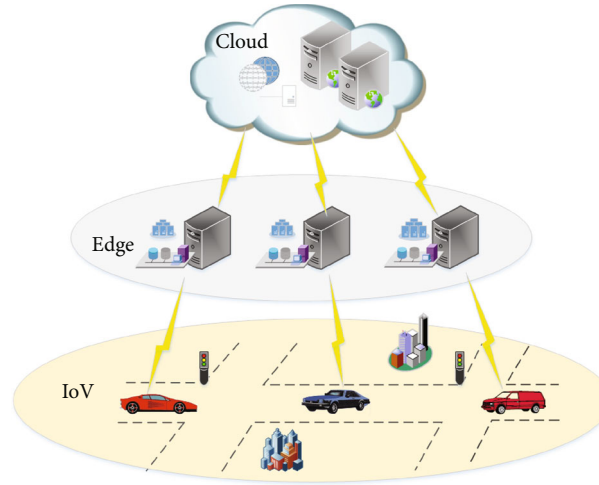

Figure 1: IoV cloud-edge-end collaborative architecture.

2. Related Work

This section includes three parts: (1) searchable encryption, (2) privacy-preserving in IoV, and (3) secure data sharing in IoV.

2.1. Searchable Encryption. With a large number of data owners outsourcing critical private data to access rich computational and storage resources at a lower cost, the SE scheme that enables encrypted data retrieval is widely studied and applied [22–24]. Song et al. [8] first introduced a symmetric search encryption (SSE) scheme in 2000 to enable a single keyword search over ciphertext. Boneh et al. [25] proposed the first Public-Key Encrypted Keyword Search (PEKS) scheme, which has a broader application scenario than the SSE scheme and can support secure data sharing among multiple data owners. Sahai and Waters [26] introduced an Attribute-Based Encryption (ABE) scheme in 2005, which supports one-to-many encryption, greatly reduces the number of keys generated, and is an effective way to achieve fine-grained access control, and since then, researchers have studied the ABE scheme extensively.

The ABE scheme mainly consists of Key-Policy ABE (KP-ABE) [27] and Ciphertext-Policy ABE (CP-ABE) [28]. Bethencourt et al. [28] first proposed the CP-ABE scheme in 2007, by using attributes to express the user's authentication credentials and tokenizing the key; the user could decrypt the ciphertext if the set of attributes hidden in the user's key matched the access policy embedded in the ciphertext by the data owner, which was more suitable for dynamic scenarios than the KP-ABE scheme that embeds the access policy in the key. Keyword search is an effective way to help users rapidly filter the data they need, based

on which researchers have conducted extensive research on Ciphertext Policy Attribute-Based Keyword Search (CP-ABKS) [29–31]. Qiu et al. [29] devised an attribute keyword search scheme that could resist keyword guessing attacks to maintain the indistinguishability of keywords and access structures. Miao et al. [30] presented an attribute-based encrypted keyword search scheme for verifiable attributes and consider the access rights of the same data based on the priority tree of the attributes. Zhang et al. [31] designed a lightweight searchable encryption protocol for industrial IoT that can provide users with connected keyword search while extending the scheme to multiauthority scenarios to efficiently generate and manage keys, but the system did not have the attribute update function and was less dynamic.

2.2. Privacy-Preserving in IoV. With the large amount of sensitive data in IoV being collected by sensors carried by vehicles, the issue of user privacy protection involved in the collection, transmission, and storage of vehicle data has received a lot of attention from researchers. Wu et al. [32] focused on vehicle anonymity and driving privacy in IoV by designing a privacy-preserving system equipped with a priori and a posteriori countermeasures for message verification thereby improving the reliability of vehicle-to-vehicle (V2V) communication. Kumar et al. [33] proposed a privacy-preserving IoV framework based on blockchain technology and built a deep learning module to detect data in the blockchain to guarantee data security. Zhou et al. [34] considered the location privacy problem of the designed EVN architecture and introduced edge computing to propose a differentiated privacy-preserving service framework. Kang et al. [35] used fog computing to achieve effective user location privacy protection and avoided high latency and cost problems. Wu et al. [36] focused on privacy leakage when computing tasks were offloaded in IoV scenarios, quantifying the potential threats when vehicle users offloaded computing tasks based on physical layer security theory.

The above schemes have effectively investigated data privacy protection in IoV, but they mainly focus on vehicle location privacy or data storage privacy and do not expand much on secure retrieval and sharing of IoV data.

2.3. Secure Data Sharing in IoV. Data sharing can further increase the value of IoV data utilization; as users are increasingly concerned about privacy protection when performing information retrieval, researchers have conducted preliminary studies on IoV data secure retrieval using existing technologies. Chen et al. [37] designed an IoV data sharing incentive mechanism based on the tamper-proof performance of blockchain to ensure the integrity of data on the chain. Cui et al. [38] designed a traceable and anonymous V2V data sharing using federated blockchain technology to track the origin of data and prevent data from being shared twice by malicious users, but the above two schemes are difficult to support the user's flexible data retrieval needs.

Several studies have improved the ABE scheme in secure retrieval of IoV data and fine-grained access control. Wang et al. [39] extended ground-based IoV scenarios to Space-Air-Ground Integrated Vehicular Networks (SAGIN); a valid keyword conversion algorithm based on a single lattice algorithm and particle encryption is proposed to achieve fuzzy retrieval, and keyword weights are calculated using dependency grammar and phrase structure tree to improve retrieval precision. Zhang et al. [40] proposed a secure retrieval scheme for IoV data based on cloud-fog collaboration, focusing on the problem of accessing sensitive data by malicious users whose attributes are revoked, proposing the concept of auditable user revocation, and giving a verifiable online/offline calculation method. Considering the problems of high computational consumption and low efficiency of serial outsourcing decryption of the ABE scheme, Feng et al. [41] introduced the edge computing to support parallel outsourcing decryption, and the designed scheme can be extended to existing ABE schemes built based on tree structure and linear secret sharing.

However, the above studies utilize the single attribute authority for complex key generation and management tasks when building a secure retrieval scheme for IoV data in combination with an ABE scheme, which is prone to the single-point performance bottleneck. In addition, for IoV scenarios, the computational complexity of the scheme should be minimized without sacrificing efficiency and security.

3. Preliminaries

3.1. Bilinear Groups. Assume that G, G_T are two multiplicative cyclic groups of order p, where p is a prime, g is a generator of G, and the bilinear mapping $e : G \times G \longrightarrow G_T$ has the following properties:

(1) *Bilinear.* $e(g^a, g^b) = e(g, g)^{ab}$, $\forall a, b \in Z_p$

(2) *Nondegeneracy.* $e(g, g) \neq 1$

(3) *Computability.* $\forall x, y \in G$, there exists a valid polynomial-time algorithm to compute the value of $e(x, y)$

3.2. Access Structure. Let $P = \{P_1, P_2, \cdots, P_n\}$ be a set of attributes, and the access structure $\Lambda \subseteq 2^P$ is monotonic. The access structure Λ is a nonempty subset of the set 2^P, the sets in Λ are authorized sets, and the sets not in Λ are unauthorized sets.

3.3. Linear Secret Sharing Scheme (LSSS). If the following conditions both hold, an LSSS over P is linear.

(1) The sharing of each attribute forms a vector on Z_P

(2) Let M $(l \times n)$ be the shared matrix of LSSS to describe the access structure Λ, the ith row is defined as $M_i (i \in [1, l])$, and the mapping function $\rho(.)$ maps each row M_i to a certain attribute $\rho(i)$. Given a randomly chosen vector $\mathbf{x} = \{s, y_1, y_2, \cdots, y_n\} \in Z_P^n$,

where s is the shared secret value, then $\mathbf{M}\mathbf{x}^T$ represents the l shares of s in LSSS, where the shared $\mathbf{M}_i$ $\mathbf{x}$ belongs to a attribute $\rho(i)$, denoted as $\lambda_i = \mathbf{M}_i\mathbf{x}$

The LSSS defined in the above way is reconfigurable: assume that $(\mathbf{M}, \rho)$ denotes the access structure Λ of the LSSS, the set of authorized users $S \in \Lambda$, and define $I = \{i, \rho(i) \in S\} \subseteq \{1, 2, \cdots, l\}$. There exists $\boldsymbol{\omega} = \{\omega_i \in Z_P\}_{i \in I}$ such that $\sum_{i \in I} \omega_i \mathbf{M}_i = (1, 0, \cdots, 0)$, and thus, $\sum_{i \in I} \omega_i \mathbf{M}_i \mathbf{x} = \sum_{i \in I} \omega_i \lambda_i = s$.

4. System Model and Definition

4.1. System Model. As shown in Figure 2, the system model involves six main participants, namely, central authority (CA), multiple attribute authorities (AAs), vehicle node (VN), multiple edge servers (ESs), cloud server (CS), and search user (SU).

(i) *Central Authority*. The CA is responsible for initializing the system and registrating multiple SUs and AAs

(ii) *Attribute Authority*. Each AA is independent of the other, and there is no intersection between the attributes managed. The AA is responsible for the generation and distribution of user keys within the domain and supports attribute updates for authorized users

(iii) *Vehicle Node*. Different kinds of sensors carried by vehicles observe the vehicle status in real-time, and the VN obtains an attribute-based access structure from AA to encrypt and upload the collected vehicle datasets to the nearest ES

(iv) *Edge Server*. The ES is mainly responsible for the following three tasks. First, it stores vehicle instant ciphertext and forwards vehicle historical ciphertext to CS. Second, it provides instant search service to SU and forwards trapdoor from SU to CS to realize historical search. Third, it provides ciphertext predecryption service to SU whose attributes satisfy the access structure

(v) *Cloud Server*. The CS provides outsourced storage and search service for the vehicle historical ciphertext. In addition, the CS sends the matching ciphertext to ES for predecryption after an accurate keyword search

(vi) *Search User*. The SU obtains the secret key from AA and wishes to freely access ciphertext resources in ES or CS without compromising privacy while reducing the computational burden of decrypting the ciphertext

In the CEABKS-MA system, the CA and multiple AAs, as fully trusted third parties, are real-time online and have sufficient computing and storage resources to perform tasks such as system initialization and key distribution. The CS and multiple ESs are "honest and curious"; they perform ciphertext storage and search services honestly but may try to obtain more private data without authorization.

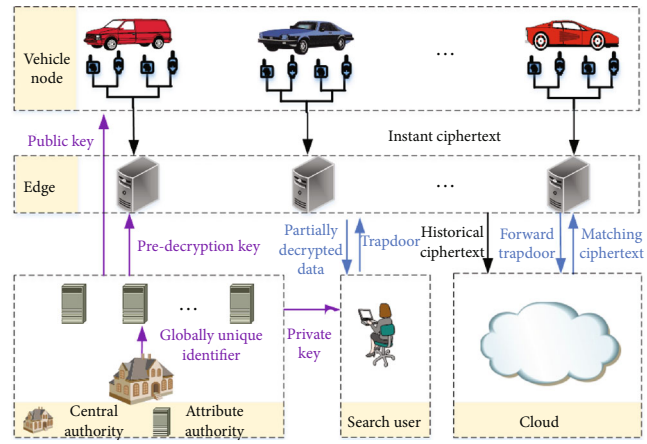

FIGURE 2: Cloud-edge-end collaborative multiauthority attribute encrypted search model in IoV.

4.2. Formal Definition. Let $S_A = \{AA_1, AA_2 \cdots, AA_N\}$ denote the set of multiple AAs, the $AA_j (j \in [1, N])$ has a set of attributes $L_j = \{att_{1,j}, att_{2,j}, \cdots, att_{U,j}\}$, and the number of attributes managed by AA_j is denoted as U_j. The proposed CEABKS-MA system includes the following polynomial-time algorithms.

(1) $CAsetup(\kappa) \longrightarrow (\mathrm{PP}, \mathrm{MSK})$. Given the security parameter κ, the CA generates the public parameters PP and the master key MSK, while generates a unique identifier *uid* for each authorized SU

(2) $AAsetup(\mathrm{PP}) \longrightarrow (\mathrm{ASK}_j, \mathrm{APK}_j)$. Given the public parameters PP, the AA_j generates an attribute public key APK_j and an attribute private key ASK_j for each set of attributes it manages

(3) $Keygen(\mathrm{PP}, \mathrm{MSK}, S_{j,uid}, \mathrm{ASK}_j) \longrightarrow \mathrm{USK}_{j,uid}$. $S_{j,uid}$ denotes the set of attributes received by SU_{uid} from AA_j, where $S_{j,uid} \subseteq L_j$; then, the AA_j generates the key $\mathrm{USK}_{j,uid}$ associated with the set of attributes for SU_{uid} and sends it to SU_{uid}

(4) $PreKeygen(z, \mathrm{USK}_{j,uid}) \longrightarrow \mathrm{PSK}_{j,uid}$. The SU_{uid} randomly selects the blind value z to send it to A A_j; then, the AA_j executes this algorithm to generate the predecryption key $\mathrm{PSK}_{j,uid}$ according to US $\mathrm{K}_{j,uid}$ and sends it to ES

(5) $Encrypt(\mathrm{PP}, \mathrm{APK}_j, E, kw, (\mathbf{M}, \rho)) \longrightarrow \mathrm{CT}$. Given the vehicle plaintext dataset E, the vehicle attribute keywords kw, and the access structure $(\mathbf{M}, \rho)$, the VN outputs the ciphertext CT, which includes

the encrypted index set I_E and the encrypted dataset C_E

(6) $Trapdoor(\mathrm{PP}, \mathrm{USK}_{j,uid}, \mathrm{kw}') \longrightarrow T_{\mathrm{kw}'}$. Given the query keyword set kw', the SU_{uid} generates the trapdoor $T_{\mathrm{kw}'}$ according to the secret key $\mathrm{USK}_{j,uid}$ and sends it to ES

(7) $Search(\mathrm{CT}, T_{\mathrm{kw}'}) \longrightarrow \{0,1\}$. Given the trapdoor $T_{\mathrm{kw}'}$ and the ciphertext CT, the ES or CS conducts the search algorithm, if the query is successful, outputs "1" and performs the ciphertext predecryption operation, otherwise, outputs "0"

(8) $EdgeDec(\mathrm{CT}, \mathrm{PSK}_{j,uid}) \longrightarrow \mathrm{CT}_{\mathrm{out}}$. Given the ciphertext CT and the predecryption key $\mathrm{PSK}_{j,uid}$, the ES outputs the partially decrypted ciphertext $\mathrm{CT}_{\mathrm{out}}$ and sends it to SU_{uid}

(9) $\mathbf{Dec}(\mathrm{CT}_{\mathrm{out}}, z) \longrightarrow sk_{E_a}$. Take the partially decrypted ciphertext $\mathrm{CT}_{\mathrm{out}}$ as input, and the SU_{uid} performs this algorithm to decrypt the ciphertext lightly using the blind value z

(10) $\mathbf{Update}(\mathrm{CT}, \mathrm{ASK}'_{i,j}, \mathrm{APK}'_{i,j}, v'_i) \longrightarrow \hat{v}'_i$. Given the ciphertext CT and the attribute update public key $\mathrm{APK}'_{i,j}$ for key and ciphertext update

4.3. Secure Model. To protect the confidentiality of vehicle data, unauthorized CS, ESs, and SUs cannot access any plaintext information. The CEABKS-MA system proposed in this paper enables the Indistinguishability of Chosen Plaintext Attack (IND-CPA) [42], as well as the Indistinguishability of Chosen Keyword Attack (IND-CKA) [43]. In this subsection, we define the following interactive game between challenger B and adversary A.

(1) IND-CPA security

Initialization. Adversary A announces a challenging access structure $\mathbf{\Lambda} *$ and sends it to challenger B.

Setup. B first runs the *Setup* algorithm, outputs the public key PK, and sends it to A.

Phase 1. A can adaptively send any attribute set S to B, but the restriction is that all submitted attribute sets cannot satisfy $\mathbf{\Lambda} *$. For each attribute set S, A executes the Keygen algorithm to output the key and sends it to B. Moreover, A can make any queries for updated key related to the canceled attribute v'_i.

Challenge. A selects two messages m_0, m_1 of equal length and sends them to B; then, B randomly selects $\kappa \in \{0,1\}$ and uses $\mathbf{\Lambda} *$ to encrypt m_κ. Finally, B returns the challenging ciphertext C_κ to A.

Phase 2. A repeats Phase 1 for other sets of attributes, but none of them satisfy $\mathbf{\Lambda} *$.

Guess. A outputs $\kappa' \in \{0,1\}$, if $\kappa' = \kappa$, then wins the security game; otherwise, it fails.

(2) IND-CKA security

Definition 1. If the advantage of winning the above game in any polynomial-time adversary is negligible, then the CEABKS-MA system is IND-CPA security.

Setup. B outputs the public key PK and sends it to A.

Phase 1. A can query O_{Keygen} and O_{Trap} for keys and trapdoors in polynomial time.

(i) O_{Keygen}. B invokes Keygen to generate the corresponding key SK and sends it to A

(ii) O_{Trap}. A submits a keyword kw' of interest, and B executes the Trap algorithm to generate the trapdoor $T_{\mathrm{kw}'}$ and sends it to A

Challenge. A selects two keywords $\mathrm{kw}_0, \mathrm{kw}_1$ with the same length, and then, B randomly selects $\kappa \in \{0,1\}$, generates index I_{kw_κ}, and returns it to A.

Phase 2. The process of Phase 2 is similar to that of Phase 1.

Guess. A outputs $\kappa' \in \{0,1\}$, if $\kappa' = \kappa$, then wins the security game; otherwise, it fails.

Definition 2. If the advantage of winning the above game in any polynomial-time adversary is negligible, then the CEABKS-MA system is IND-CKA security.

5. The Proposed CEABKS-MA System

5.1. Construction of CEABKS-MA System

5.1.1. System Initialization. Assume that $H : \{0,1\}* \longrightarrow Z_p$ is a one-way hash function, and $e : G \times G \longrightarrow G_T$ is chosen as a bilinear mapping, where G and G_T are p-order cyclic groups whose generators are g and g_T, respectively. The initialization process is divided into two stages, which are described in detail as follows.

(i) $CAsetup(\kappa)$. The CA executes the algorithm using the security parameter κ, obtaining the global bilinear parameter $GP = (e, g, G_T, G)$, and then randomly selects $a_0, a_1 \in Z_p^*$ to compute $Y = e(g,g)^{a_0}$, finally obtains the public parameter $PP = \{GP, g^{a_0}, g^{a_1}, Y\}$ and the master key $\mathrm{MSK} = \{a_0, a_1\}$

(ii) $AAsetup(\mathrm{PP})$. For each attribute $\mathrm{Att}_{i,j} \in L_j (i \in [1, U_j])$, the AA_j picks a random element $\alpha_i \in Z_p$ and computes $h_i = g^{\alpha_i}$, then randomly chooses $v_i \in Z_p$ to get the attribute version key $\mathrm{APK}_{i,j} = g^{v_i}$, $\mathrm{ASK}_{i,j} = v_i$. Finally, the AA_j gets the attribute private key $\mathrm{ASK}_j = \{\{\alpha_i\}, \{v_i\}\}_{i\in[1,U_j], j\in[1,N]}$ and the attribute public key $\mathrm{APK}_j = \{\{h_i\}, \{\mathrm{APK}_{i,j}\}\}_{i\in[1,U_j], j\in[1,N]}$

5.1.2. Key Generation

(i) *Keygen*. The AA_j computes $K' = g^{a_0 a_1}$, and for each attribute $\tau \in S_{j,uid}$, picks a random value $t \in Z_p$ and computes $K_1 = g^{a_0} g^{a_1 t}$, $K_2 = g^t$, finally constructs a

private key $\mathrm{USK}_{j,uid} = \{K', K_1, K_2\}$ and sends it to SU_{uid}

(ii) *PreKeygen*. The SU_{uid} selects a random value $z \in Z_p$ and sends it to the AA_j, the AA_j computes $K'_1 = K_1^z$, $K'_2 = K_2^z$, and $K'_\tau = h_\tau^{tz/v_\tau}$, then constructs a predecryption key $\mathrm{PSK}_{j,uid} = \{K'_1, K'_2, \{K'_\tau\}_{\tau \in S_{j,uid}, j \in S_A}\}$ and sends it to ES

5.1.3. Ciphertext and Encrypted Index Generation. In the actual IoV scenario, different types of sensors carried by vehicles collect the corresponding vehicle attribute data separately. For the different attribute states of vehicles monitored by different sensors deployed on the same vehicle, the CEABKS-MA system can achieve a fine-grained keyword search for the specified vehicle attributes. Given the vehicle attribute dataset $E = \{E_a\}$ and a keyword dictionary $\mathrm{KW} = \{\mathrm{kw}\}$, the VN uses the key sk_{E_a} to encrypt the data of each attribute of the vehicle $E_a \in E$ and defines the encrypted vehicle attribute dataset as $C_E = \{\mathrm{Enc}_{sk_{E_a}}(E_a)\}$; the symmetric key sk_{E_a} is protected by a specified access structure (M, ρ), where $\mathbf{M}$ is the matrix of $n \times l$; ρ is a function that associates rows of $\mathbf{M}$ to attributes. The specific encryption process is divided into vehicle attribute data encryption and vehicle attribute index encryption, as follows.

(i) *Encrypt*(CT_E). The VN chooses two random vectors $\mathbf{x} = \{s, y_1, y_2, \cdots, y_n\} \in Z_p$ and $r = \{r_1, \cdots, r_l\} \in Z_p$, where s is the secret sharing value, and computes $\lambda_i = \mathbf{M}_i \mathbf{x}$, where $i \in [1, l]$. Then, for $\forall i \in [1, l]$, the VN computes $C_{i,1} = g^{a_1 \lambda_i} h_{\rho(i)}^{-r_i}$, $C_{i,2} = g^{r_i v_{\rho(i)}}$, $C' = g^s$, and $C_{E_a} = sk_{E_a} \cdot e(g, g)^{a_0 s}$ and outputs the vehicle attribute ciphertext $\mathrm{CT}_{E_a} = \{C', C_{E_a}, \{C_{i,1}, C_{i,2}\}_{i \in [1,l]}\}$, so as to get the vehicle ciphertext set $\mathrm{CT}_E = \{\mathrm{CT}_{E_a}\}$

(ii) *Encrypt*(I_E). The VN extracts keywords $\mathrm{kw} \in \mathrm{KW}$ from different attribute dataset $E_a \in E$ and constructs an attribute encrypted index I_{E_a} based on the keywords in each E_a. Then, the VN selects a random element $\pi \in Z_p$, for $\forall i \in [1, l]$, computes $I_0 = g^{a_1 \pi}$ and $I_{1,i} = g^{a_0(s+\pi)} h_{\rho(i)}^{\pi H(kw)}$ and outputs the vehicle attribute encrypted index $I_{E_a} = \{I_0, \{I_{1,i}\}_{i \in [1,l]}\}$, so as to get the vehicle encrypted index set $I_E = \{I_{E_a}\}$

(iii) The VN uploads the vehicle ciphertext to ES periodically, and after the ciphertext expires (i.e., the VN uploads a new round of ciphertext), the ES uploads this vehicle historical ciphertext to CS

5.1.4. Trapdoor Generation. If the SU_{uid} uses his key and keyword set to generate a trapdoor $T_{\mathrm{kw}'}$ to search an attribute status of the vehicle that contains the query keyword kw', as follows.

(i) *Trapdoor*. The SU_{uid} randomly selects $\mu \in Z_p^*$ and computes $T_1 = g^{a_1 \mu}$ and $T_2 = (K')^\mu$. Then, according to the query keyword kw', for each attribute $\tau \in S_{j,uid}$, the SU computes $T_0 = g^{a_0 \mu} \prod_{\tau \in S_{j,uid}} h_\tau^{\mu H(\mathrm{kw}')}$, finally gets the search trapdoor $T_{\mathrm{kw}'} = \{T_0, T_1, T_2\}$ and sends it to ES

5.1.5. Search and Predecryption. After receiving the trapdoor and the attribute set $S_{j,uid}$ from SU_{uid}, it is mainly divided into two processes: *Search* and *EdgeDec*.

(i) *Search*. The CS or ES first verifies whether the attribute set of SU_{uid} embedded in the trapdoor $T_{\mathrm{kw}'}$ can satisfy the access structure (M, ρ) of the ciphertext CT and stops the search operation if it does not match; otherwise, the keyword search algorithm is executed to match the trapdoor $T_{\mathrm{kw}'}$ and the index set I_E, as shown as follows:

$$e(I_0, T_0) e\left(C', T_2\right) = e\left(\prod_{\rho(i) \in S_{j,uid}} I_{1,i}, T_1\right), \tag{1}$$

Correctness verification is as follows:

$$\begin{aligned} \xi_1 &= e\left(I_0, \prod_{\tau \in S_{j,uid}} T_0\right) e\left(C', T_2\right) \\ &= e\left(g^{a_1 \pi}, g^{a_0 \mu} \prod_{\tau \in S_{j,uid}} h_\tau^{\mu H\left(kw'\right)}\right) e\left(g^s, g^{a_0 a_1 \mu}\right) \\ &= e(g, g)^{a_0 a_1 \mu \pi} e\left(g, \prod_{\tau \in S_{j,uid}} h_\tau\right)^{a_1 \mu \pi H\left(kw'\right)} e(g, g)^{a_0 a_1 \mu s}, \end{aligned} \tag{2}$$

$$\begin{aligned} \xi_2 &= e\left(\prod_{\rho(i) \in S_{j,uid}} I_{1,i}, T_1\right) \\ &= e\left(g^{a_0(s+\pi)} \prod_{\rho(i) \in S_{j,uid}} h_{\rho(i)}^{\pi H(kw)}, g^{a_1 \mu}\right) \\ &= e(g, g)^{a_0 a_1 \mu s} e(g, g)^{a_0 a_1 \mu \pi} e\left(g, \prod_{\rho(i) = \tau \in S_{j,uid}} h_{\rho(i)}\right)^{a_1 \mu \pi H(kw)}. \end{aligned} \tag{3}$$

Obviously, when $\mathrm{kw}' = \mathrm{kw}$, there is $\xi_1 = \xi_2$; that is, the keyword search algorithm is successful and outputs "1," otherwise, outputs "0."

(ii) *EdgeDec*. After the keyword search is successful, the ES will perform the ciphertext predecryption operation for SU_{uid}. Define $I \subset \{1, 2, \cdots, l\}$, expressed as

$I = \{i, \rho(i) \in S_{j,uid}\}$; there must be a set of constants $\{w_i \in Z_p\}_{i\in I}$ makes $\sum_{i\in I} w_i \lambda_i = s$, and calculates the following:

$$\begin{aligned}\zeta &= \frac{e\left(K'_1, C'\right)}{\prod_{i\in I}\left[e\left(C_{i,1}, K'_2\right)\cdot e\left(C_{i,2}, K_{\rho(i)}\right)\right]^{w_i}} \\ &= \frac{e(g^{a_0 z} g^{a_1 tz}, g^s)}{\sum_{i\in I}\left[g^{a_1\lambda_i}\cdot h_{\rho(i)}^{-r_i}, g^{tz}\right) e\left(g^{r_i v_{\rho(i)}}, h_{\rho(i)}^{tz/v_i}\right)\Big]^{w_i}} \\ &= \frac{e(g,g)^{a_0 zs} e(g,g)^{a_1 tzs}}{\sum_{i\in I}\left(e(g,g)^{a_1 tz\lambda_i}\right)^{w_i}} = e(g,g)^{a_0 zs}.\end{aligned} \tag{4}$$

The ES constructs partially decrypted ciphertext $\mathrm{CT}_{\mathrm{out}} = \{C_{\mathrm{kw}'}, \zeta\}$ and returns it to SU_{uid}.

5.1.6. User Decryption

(i) After receiving the partially decrypted ciphertext, the SU uses the blind value z to compute $C_{\mathrm{kw}'}/\zeta$ to obtain the symmetric key sk_{E_a} and then uses sk_{E_a} to obtain the plaintext vehicle data E_a

5.2. Attribute Revocation and Update. The access right change of SU requires the update of their attributes to avoid malicious users from using expired keys to access unauthorized information. Each AA in the CEABKS-MA system manages a disjoint set of attribute collections and performs attribute update operations only for users in the domain, effectively spreading the computational and storage burden of the CA and obtaining higher efficiency.

When there are some attributes to be updated, the AA_j first updates the attribute version key $\mathrm{ASK}_{i,j}$, $\mathrm{APK}_{i,j}$ and then generates the transformation key to update SU's key and the vehicle ciphertext stored in ES or CS. Moreover, the CEABKS-MA system only updates a small portion of the attribute-related key and ciphertext; the attribute update algorithm is as follows.

(i) If the attribute att'_i of SU managed by AA_j is revoked, the AA_j inputs $\mathrm{ASK}_{i,j}, \mathrm{APK}_{i,j}$, and the revoked attribute att'_i randomly chooses a new value $\widehat{v}'_i \in Z_p (\widehat{v}'_i \neq v'_i)$ and computes the updated attribute version key as $\mathrm{ASK}'_{i,j} = \widehat{v}'_i / v'_i$, $\mathrm{APK}'_{i,j} = \left(g^{v'_i}\right)^{\mathrm{ASK}'_{i,j}} = g^{\widehat{v}'_i}$. Finally, the AA_j sends $\mathrm{ASK}'_{i,j}$ to ES or CS

(ii) *Key Update*. The AA_j informs SU that has the attribute att'_i and has not been revoked to upload the relevant part of the key component with the revoked attribute to AA_j for updating. After receiving the data uploaded by SU, the AA_j computes $\widehat{K}'_i = (K'_i)^{\mathrm{ASK}'^{-1}_{i,j}} = \left(h_i^{zt/v'_i}\right)^{v'_i/\widehat{v}'_i} = h_i^{zt/\widehat{v}'_i}$ and returns it to SU whose attributes have not been revoked

(iii) *Ciphertext Update*. When the attribute att'_i of SU is revoked, the AA_j needs to update the ciphertext synchronously. Due to the limited computing resources of VN, updating the attribute ciphertext $\widehat{C}_{i,2} = (C_{i,2})^{\mathrm{ASK}'_{i,j}} = \left(g^{r_i v'_i}\right)^{\widehat{v}'_i/v'_i} = g^{r_i \widehat{v}'_i}$ associated with attribute att'_i on ES or CS

5.3. Online/Offline Encryption

5.3.1. Ciphertext Online/Offline Generation. To avoid the heavy burden of encrypting computation as well as to improve the efficiency of encryption, the CEABKS-MA system is extended to support ciphertext online/offline generation, as follows.

(i) *Offline Encryption*. Let the maximum number of lines in the access structure $(\mathbf{M}, \rho)$ embedded in the vehicle ciphertext be Θ. The VN chooses random elements $s, \pi, \lambda'_i, r_i \in Z_p$ and computes $C'_{i,1} = g^{a_1\lambda'_i}$, $C'^{\,}_{i,1} = h_{\rho(i)}^{-r_i}$, $C_{i,2} = g^{r_i v_{\rho(i)}}$, $C'_{E_a} = e(g,g)^{a_0 s}$, $C' = g^s$, $I_0 = g^{a_1\pi}$, $I'_1 = g^{a_0(s+\pi)}$, and $I''_{1,i} = h_{\rho(i)}^{\pi}$. Finally, the VN generates the offline vehicle attribute ciphertext $\mathrm{CT}'_{E_a} = \{C', C'_{E_a}, \{C'_{i,1}, C'^{\,}_{i,1}, C_{i,2}\}_{i\in[1,\Theta]}\}$ and the offline vehicle attribute encrypted index $I'_{E_a} = \{I_0, I'_1, I''_{1,i}\}_{i\in[1,\Phi]}$

(ii) *Online Encryption*. The VN selects a random vector $\mathbf{x} = \{s, y_1, y_2, \cdots, y_n\} \in Z_p$, s as the secret shared value of the access structure $(\mathbf{M}, \rho)$ and computes $\lambda_i = \mathbf{M}_i \mathbf{x}$, $i \in [1, l]$. Then, the VN computes $C'_i = \lambda_i - \lambda'_i$ and $C_{E_a} = sk_{E_a} \cdot C'_{E_a}$ and gets the complete vehicle attribute ciphertext $\mathrm{CT}_{E_a} = \{C_{E_a}, C', C'_i, \{C'_{i,1}, C'^{\,}_{i,1}, C_{i,2}\}_{i\in[1,l]}\}$. Finally, for $\forall i \in [1, l]$, the VN computes $I_{1,i} = I'_1 \cdot I''^{\,H(\mathrm{kw})}_{1,i}$ and gets the complete vehicle attribute encrypted index $I_{E_a} = \{I_0, \{I_{1,i}\}_{i\in[1,l]}\}$

5.3.2. Trapdoor Online/Offline Generation. Similarly, the trapdoor generation part is divided into the online/offline phase to improve the computation efficiency of SU.

(i) *Offline Generation*. The SU randomly selects $\mu \in Z_p^*$ and computes $T_0{}' = g^{a_0\mu}$, $T_1 = g^{a_1\mu}$, and $T_2 = (K')^{\mu}$, then gets the offline part of the trapdoor $T_{\mathrm{off}} = \{T'_0, T_1, T_2\}$. The SU saves it to avoid duplicate operations during the search

(ii) *Online Generation*. Based on the vehicle attribute keyword kw', for each attribute $\tau \in S_{j,uid}$, the SU

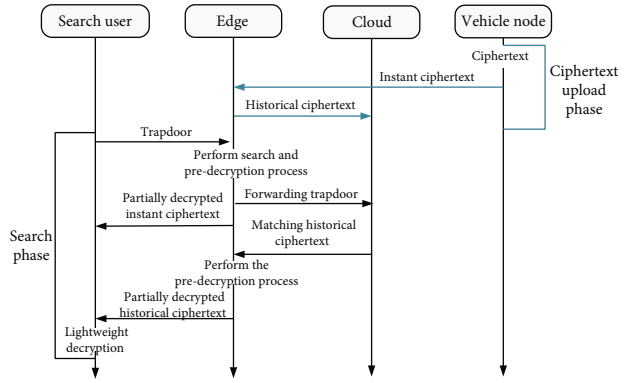

Figure 3: Cloud-edge-end collaborative search method.

Table 1: Functional comparison of different system.

System	HP-CPABKS [29]	VKSE [30]	LSABE-MA [31]	CEABKS-MA
Keyword search	√	√	√	√
Attribute revocation	×	×	×	√
Multi-authority	×	×	√	√
High efficiency	×	×	×	√
Access structure	LSSS	Tree	LSSS	LSSS

Table 2: Theoretical computation cost comparison.

System	CEABKS-MA	HP-CPABKS [29]
Keygen	$(\lvert S\rvert + 5)E$	$(2\lvert U\rvert + 1)E + E_T$
Trapdoor	$(\lvert S\rvert + 2)E$	$(2\lvert U\rvert + 1)E$
Encrypt	$(3\lvert U_j\rvert + 3)E + E_T$	$(2\lvert U\rvert + 1)E + E_T$
Search	$3P$	$(2\lvert U\rvert + 1)P + E_T$

computes $T_0 = T'_0 \prod_{\tau \in S_{j,uid}} h_\tau^{\mu H(\mathrm{kw}')}$ and gets the online trapdoor $T_{\mathrm{on}} = \{T'_0\}$, then gets the final trapdoor $T_{\mathrm{kw}'} = \{T_{\mathrm{off}}, T_{\mathrm{on}}\}$

(iii) *Predecryption Phase.* In the predecryption phase, the CS or ES first computes $C''_i = g^{a_1 C'_i} = g^{a_1(\lambda_i - \lambda'_i)}$ and $C_{i,1} = C'_{i,1} C''_i C''_{i,1} = g^{a_1 \lambda_i} h_{\rho(i)}^{-r_i}$, which can be predecrypted by using the predecryption formula (4)

5.4. Cloud-Edge-End Collaborative Search Method. This paper designs a cloud-edge-end collaborative search method to provide a more efficient and flexible search while reducing user burden. The specific search process is shown in Figure 3.

The object task in the proposed search method has changed compared to the cloud-based search method. In the ciphertext upload phase, the vehicle carries sensors to monitor the vehicle state in real-time, and the VN encrypts the vehicle data and uploads the ciphertext to the nearby ES periodically for reducing communication cost and latency caused by long-distance communication toward CS. And after the VN forwards a new round of the ciphertext, the ES uploads the historical ciphertext to CS to reduce the computation and storage burden. In the search phase, the SU only needs to send the trapdoor to ES for instant and historical search; at the same time, the corresponding ciphertext after a successful query is finally returned to SU after predecryption by ES, and the SU only needs to perform marginal decryption operation to decrypt it.

6. Safety and Performance Simulation Validation

6.1. Security Analysis. The CEABKS-MA system proposed in this paper can achieve IND-CPA security and IND-CKA security presented in Section 4.3 and is analyzed in detail as follows.

Theorem 3. *Under the assumption that the Decisional q-parallel Bilinear Diffie-Hellman Exponent (BDHE) assumption [44] holds, that the advantage of all polynomial-time opponents who can win the IND-CPA game can be ignored.*

Proof. Assume that adversary A can break the CEABKS-MA system by a nonnegligible advantage δ. A chooses a challenging matrix $\mathbf{M}*$, and then, B handles the q-DBDHE problem as follows.

Setup. Given a q-DBDHE challenge instance (ϕ, R), B first chooses $a_0 \in Z_p$ and sets $a_0 = a' + {a_1}^{q+1}$; then, B defines the public key component $Y = e(g,g)^{a_0} = e(g^{a_1}, g^{{a_1}^q}) e(g,g)^{a'}$. B chooses a random value $v_i \in Z_p$ for each $i \in U_j$ and sets $\mathrm{APK}_{i,j} = g^{v_i}$. To simulate the group elements h_i, B picks a random element $\beta_i \in Z_p$ for each $i \in U_j$. Let $\rho*(\iota) = i$, then B sets h_i as follows:

$$h_i = g^{\beta_i} \prod_{\iota \in \Phi} g^{a_1 \mathbf{M}^*_{i,1}/b_\iota} g^{{a_1}^2 \mathbf{M}^*_{i,2}/b_\iota} \cdots g^{{a_1}^{n^*} \mathbf{M}^*_{i,n^*}/b_\iota}, \tag{5}$$

where Φ denotes the set of indices ι. If $\Phi = \varnothing$, B sets $h_i = g^{\beta_i}$, and the values of h_i are randomly distributed due to g^{β_i}.

Phase 1. In this phase, B needs to answer A's key queries. Assume that A provides an attribute set S that do not satisfy $\mathbf{M}*$, and B chooses a vector $\mathbf{x}* = \{x_1^*, \cdots, x_{n^*}^*\}$ such that $x_1^* = -1$ for all $\iota(\rho*(\iota) \in S)$ have $\mathbf{x}* \cdot \mathbf{M}_\iota^* = 0$. Then, B randomly chooses an element $\vartheta \in Z_p$ and defines t as follows:

$$t = \vartheta + x_1^* {a_1}^q + x_2^* {a_1}^{q-1} + \cdots + x_{n^*}^* {a_1}^{q-n^*+1}. \tag{6}$$

Then, B computes K'_2:

$$K'_2 = g^{\vartheta z} \prod_{\iota=1}^{n^*} g^{\mathbf{x}^*_\iota z {a_1}^{q+1-\iota}} = g^{tz}. \tag{7}$$

Based on the definition of t above, it can be inferred that $g^{a_1 t}$ contains $g^{a_1{}^{q+1}}$ which can be cancelled by g^{a_0}. Thus, B computes K'_1 as follows:

$$K'_1 = g^{a'z} g^{a_1 \vartheta z} \prod_{\iota \in [2,n*]} g^{v_\iota a_1{}^{q+2-\iota}} = g^{a_0 z} g^{a_1 t z}. \tag{8}$$

For each attribute $\tau \in S$, B defines K'_τ if $\rho*(\iota) \neq \tau$ sets $K'_\tau = (K'_2)^{g^{\beta_\tau}/v_\tau}$. Under this condition, B cannot simulate K'_τ for the attribute $\tau \in S$ in $\mathbf{M}*$, since K'_τ contains the term $g^{a_1{}^{q+1}/b_\iota}$. If there exists a set $\Phi = \{\iota\}$ such that $\rho*(\iota) = \tau$ and B computes K'_τ as follows:

$$\begin{aligned} K'_\tau = K'_2{}^{\beta_\tau/v_\tau} \prod_{\iota \in \Phi} \prod_{k \in [1,n^*]} \left(g^{a_1{}^k \vartheta/b_\iota}\right)^{\mathbf{M}^*_{i,k}/v_\tau} \\ \cdot \prod_{\iota \in \Phi} \prod_{k \in [1,n^*]} \prod_{k' \in [1,n^*]} \left(g^{a_1{}^{q+1+k-k'}/b_\iota}\right)^{\mathbf{M}^*_{i,j}/v_\tau}. \end{aligned} \tag{9}$$

A sends a revoked attribute att'_i to perform an updated attribute version key query. B randomly selects a new value $\hat{v}'_i \in Z_p (\hat{v}'_i \neq v'_i)$ and computes the updated attribute version key as $\mathrm{ASK}'_{i,j} = \hat{v}'_i / v'_i$ and returns it as A.

Challenge. A submits two challenging messages m_0, m_1 to B with corresponding encryption keys sk_0, sk_1, and then, B randomly selects $\kappa \in \{0,1\}$ and computes $C_{E_\kappa} = sk_{E_\kappa} \cdot e(g,g)^{a_0 s}$, $C' = g^s$. However, since the ciphertext component $C_{i,1}$ contains some terms that should be removed, it is difficult to simulate $C_{i,1}$, where $i \in [1, l^*]$. To solve this problem, B randomly chooses $y^*_2, \cdots, y^*_{n^*} \in Z_p$ and shares the secret s as follows:

$$\mathbf{x} = \left(s, sa_1 + y^*_2, sa_1^2 + y^*_3, \cdots, sa_1^{n^*-1} + y^*_{n^*}\right). \tag{10}$$

Furthermore, B chooses random elements $r^*_2, \cdots, r^*_{n^*} \in Z_p$. Let Q_i be the set of all $\rho(i) = \rho(k')$ satisfying $i \in [1, l^*]$. Finally, B outputs $C_{i,1}, C_{i,2}$ as follows:

$$\begin{aligned} C_{i,1} &= h^{r^*_i}_{\rho^*(i)} \prod_{k \in [2,n^*]} (g^{a_1})^{\mathbf{M}^*_{i,k} y^*_{k^*}} \left(g^{b_i s}\right)^{-\beta_{\rho^*(i)}} \cdot \prod_{k' \in \Omega_i} \prod_{k \in [1,n^*]} \left(g^{a_1^k s b_i / b_{k'}}\right)^{\mathbf{M}^*_{k',k}}, \\ C_{i,2} &= g^{\left(-r^*_i - s b_i\right) v_{\rho^*(i)}}. \end{aligned} \tag{11}$$

Phase 2. Phase 2 has the same process as Phase 1.

Guess. A returns a guess bit $\kappa' \in \{0,1\}$, if $\kappa' = \kappa$; B returns "0" indicating that $R = e(g,g)^{a_1{}^{q+1} s}$; otherwise, B returns "1" indicating that R is a randomly chosen element of the group G_T. When R is a tuple, B returns a perfect simulation, which then yields $\Pr[B(\phi, R = e(g,g)^{a_1{}^{q+1} s}) = 0] = 1/2 + \delta$. When R is a random element in the group G_T and the encryption key sk_κ is completely hidden from A, then one obtains $\Pr[B(\phi, R) = 0] = 1/2$. Thus, B simulates the above security game with a nonnegligible advantage. This completes the proof of Theorem 3. □

Theorem 4. *Based on a given one-way hash function H, the CEABKS-MA system prevents chosen keyword attacks.*

Selecting a random value $d \in Z_p$, the advantage of adversary A in distinguishing between g^d and $g^{a_0(s+\pi)} h^{\pi H(kw_0)}_{\rho(i)}$ is the same as the advantage of distinguishing between g^d and $g^{a_0(s+\pi)} h^{\pi H(kw_1)}_{\rho(i)}$ with the same advantage. Assume that A can distinguish between g^d and $g^{a_0(s+\pi)}$, and the defined secure interactive game is as follows.

Proof. Setup. B randomly selects $a_0, a_1 \in Z_p$ and returns the public key $\mathrm{PK} = (g, g^{a_0}, g^{a_1})$ to A.

Phase 1. A can query O_{Keygen} and O_{Trap} for keys and trapdoors in polynomial time.

(i) O_{Keygen}. B computes $K' = g^{a_0 a_1}$ and sends K' to A

(ii) O_{Trap}. B randomly selects $\mu \in Z_p$ and computes $T_0 = g^{a_0 \mu} h^{\mu H(\mathrm{kw}')}_\tau$, $T_1 = g^{a_1 \mu}$, and $T_2 = g^{a_0 a_1 \mu}$ according to query keyword kw', which gives the trapdoor $T_{\mathrm{kw}'} = \{T_0, T_1, T_2\}$

Challenge. A inputs two keywords of the same length k $\mathrm{w}_0, \mathrm{kw}_1$. B selects $s, \pi \in Z_p$ and picks $\kappa \in \{0,1\}$. If $\kappa = 0$, B sets $I_0 = g^{a_1 \pi}$, $I_1 = g^d$, and $C' = g^s$, otherwise, sets $I_0 = g^{a_1 \pi}$, $I_1 = g^{a_0(s+\pi)}$, and $C' = g^s$.

Phase 2. A performs a query similar to Phase 1 but restricts $\mathrm{kw} \neq \mathrm{kw}_0, \mathrm{kw}_1$.

Assume that $v \in Z_p$ and if A can construct $e(g,g)^{v a_0(s+\pi)}$ using the term g^v returned by the query, then A can distinguish between g^d and $g^{a_0(s+\pi)}$. Thus, it needs to be shown that A can only use the term g^v to construct $e(g,g)^{v a_0(s+\pi)}$ by a negligible advantage.

Let $G_1 = \{\phi_1(\eta) | \eta \in Z_p\}$, $G_T = \{\phi_2(\eta) | \eta \in Z_p\}$, where ϕ_1 and ϕ_2 are two introjection functions mapped from Z_p to a set with p^3 elements. In the mapping between ϕ_1 and ϕ_2, the advantage of adversary A in distinguishing elements is negligible, so it is only necessary to consider the probability of adversary A in constructing $e(g,g)^{v a_0(s+\pi)}$ using g^v.

If A want to get $e(g,g)^{v a_0(s+\pi)}$ from g^v, since only $a_1 \pi$ contains π, v must contain a_1 to get $e(g,g)^{v a_0(s+\pi)}$. A will try to construct $e(g,g)^{v' a_0(s+\pi)}$ based on $v' = v a_1$. However, A also needs to get $v' a_0 a_1 s$ containing the term $a_0 a_1$ and the secret value s. Since only B has the primary key a_1, A cannot obtain $e(g,g)^{v' a_0(s+\pi)}$.

Thus, it can be concluded that adversary A cannot distinguish $g^{a_0(s+\pi)} h^{\pi H(kw_0)}_{\rho(i)}$ and $g^{a_0(s+\pi)} h^{\pi H(kw_1)}_{\rho(i)}$. That is, the CEABKS-MA system is secure in the chosen keyword attack game, which completes the proof of Theorem 4. □

Table 3: Theoretical storage cost comparison.

System	CEABKS-MA	HP-CPABKS [29]
Setup	$(2\|U_j\|+1)Z_p + (2\|U_j\|+2)\|G\| + \|G_T\|$	$(\|U\|+2)\|Z_p\| + (\|U\|+1)\|G\| + \|G_T\|$
Keygen	$(\|S\|+5)\|G\|$	$(2\|U\|+1)\|G\| + (\|U\|+2)\|Z_p\|$
Trapdoor	$3\|G\|$	$(2\|U\|+1)\|G\| + 2\|Z_p\|$
Encrypt	$(3\|U_j\|+3)\|G\| + \|G_T\|$	$(2\|U\|+1)\|G\| + (\|U\|+1)\|Z_p\|$

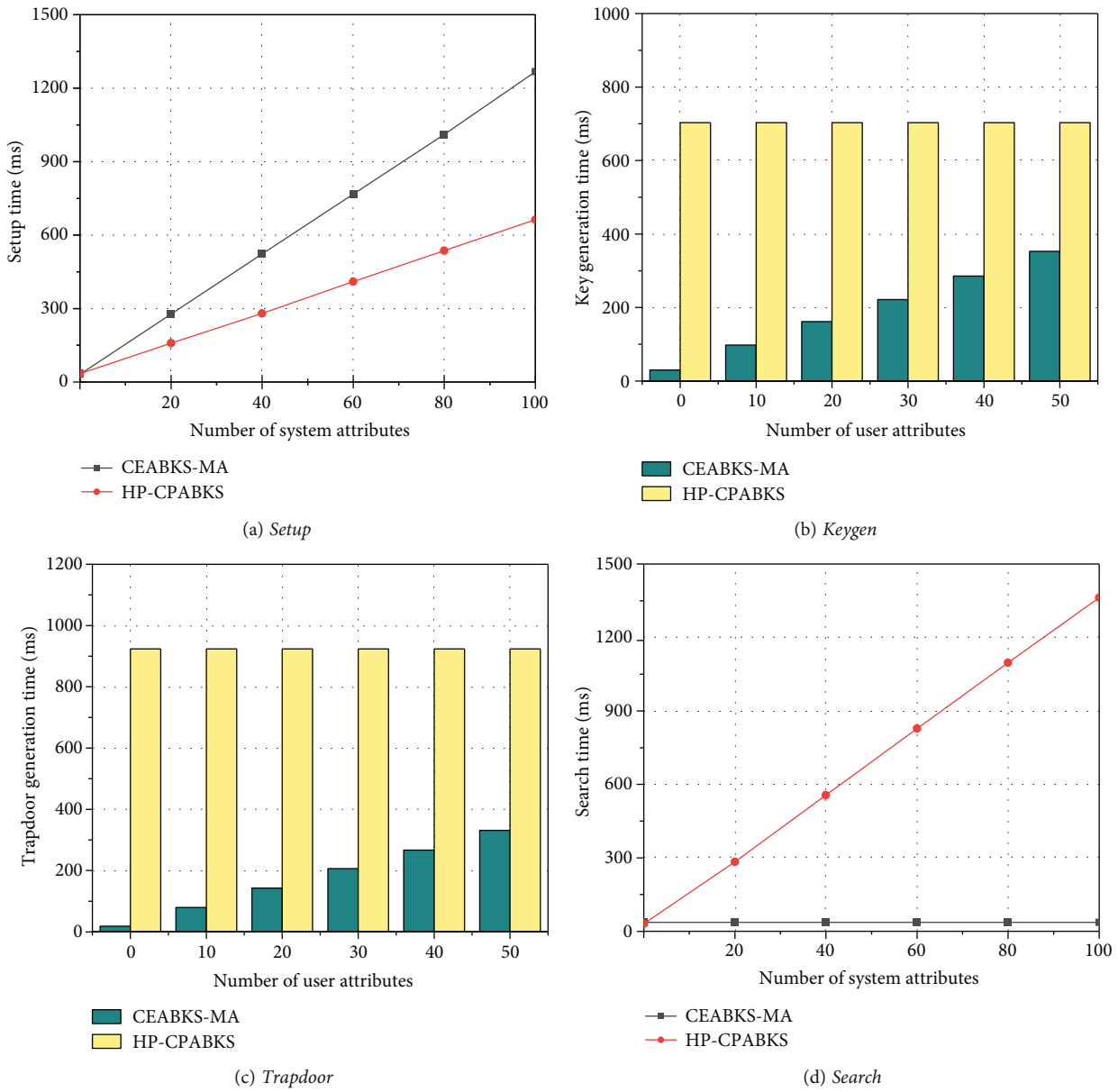

Figure 4: Computation cost comparison.

In addition, the CEABKS-MA system can resist collusion attacks by users and achieve the security of user key. (1) The CEABKS-MA system prevents user collusion attacks by assigning a global identifier *uid* to each DU. In Keygen, the key component is associated with a random value t, so it is difficult for a malicious user to isolate the t value from a given key to perform collusion queries in the absence of a random value t. (2) The search user uses a random value μ to blind the key when performing queries to ensure the security and confidentiality of the user's key.

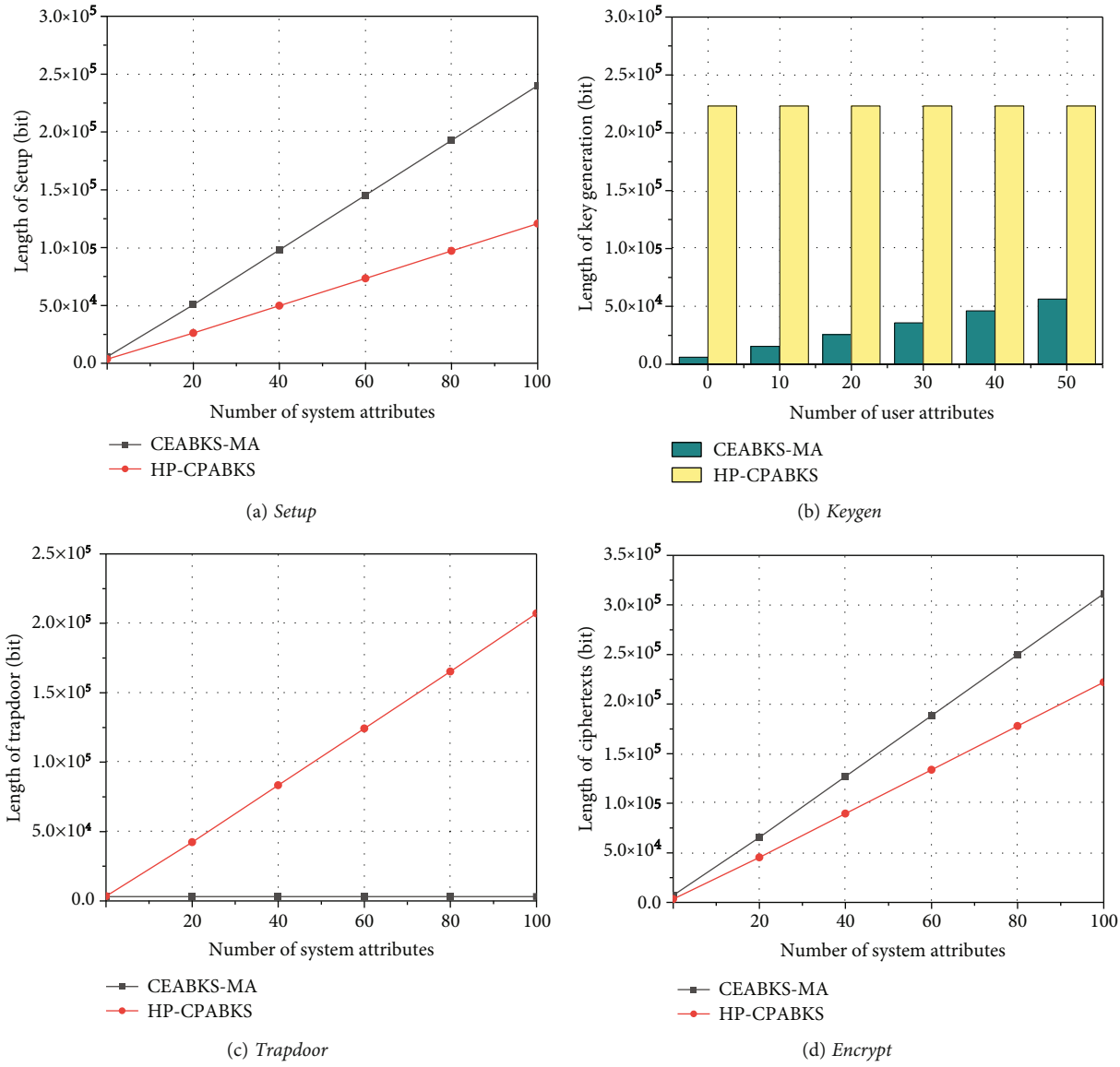

Figure 5: Storage cost comparison.

6.2. Performance Analysis. The CEABKS-MA system implements fine-grained keyword search, multiauthority structure, and attribute update and has high efficiency in both key and trapdoor generation as well as search and decryption phases. Table 1 shows a functional comparison between the CEABKS-MA system and other existing systems [29–31].

The theoretical computation and storage costs of the CEABKS-MA system and the existing scheme [29] are analyzed, as shown in Tables 2 and 3, respectively. For the computation costs in Table 2, we mainly consider several more time-consuming operations, namely, bilinear pairing operation P and exponential operation E or E_T in group G or G_T. The number of system attributes is denoted as $|U_j|$ and $|U|$ for the CEABKS-MA system and HP-CPABKS system, respectively, and the number of user attributes is denoted as $|S|$. Since the CEABKS-MA system uses a distributed key distribution structure, $|S| \ll |U_j| \ll |U|$ in practice, the CEABKS-MA system consumes less time than the HP-CPABKS system in *Keygen* and *Trapdoor*. The computation cost of the CEABKS-MA system in *Encrypt* will be higher than that of the HP-CPABKS system when setting $|U_j| = |U|$, but the online/offline encryption method is extended to the proposed system, and the ciphertext generation is a one-time operation. In *Search*, the computation cost of the CEABKS-MA system is constant, and the search efficiency is much higher than that of the HP-CPABKS system.

For the storage costs in Table 3, element lengths in G, G_T, Z_p are defined as $|G|, |G_T|, |Z_p|$, respectively. When $|U_j| = |U|$, the storage cost of the CEABKS-MA system in *Setup* is higher due to the added attribute update

function. Similar to the computation cost analysis, the storage cost of the CEABKS-MA system is much lower than that of the HP-CPABKS system in *Keygen* and *Trapdoor*, and the storage cost in *Trapdoor* is constant, which is more suitable for resource-constrained devices.

To verify the above theoretical analysis, we present an experimental analysis of the computation efficiency and storage consumption of the CEABKS-MA system and the HP-CPABKS system. The experimental simulation is Windows 10, Intel(R) Core(TM) i3-8100 CPU@3.60 GHz. The programming language is C and parsing-based cryptography (PBC) libraries. The parameters related to computation and storage costs are set as $|G| = |G_T| = 1024$ bits, $|Z_p| = 160$ bits, $|S| \in [1, 50]$, and $|U_j| = |U| \in [1,100]$.

Figure 4 shows the actual computation time comparison of different systems in each phase; in Figure 4(a), the computation cost of both systems in *Setup* increases with the expanding number of system attributes, and the CEABKS-MA system costs slightly more time than the HP-CPABKS system, which is consistent with the theoretical analysis, but note that $|U_j| < |U|$ in practice. The number of system attributes in Figures 4(b) and 4(c) is fixed at $|U| = 50$; it can be seen that the time consumption of the CEABKS-MA system in *Keygen* and *Trapdoor* increases linearly with the number of user attributes but is still much lower than that of the HP-CPABKS system, and $|S| \ll |U|$, so the CEABKS-MA system has higher efficiency and application value for search users with limited computational resources. Figure 4(d) shows the comparative analysis of search time, which is constant and much lower than that of the HP-CPABKS system.

Figure 5 shows the actual storage cost comparison of different systems in each phase, where Figures 5(a)–5(c) are consistent with the reasons analyzed in Figures 4(a)–4(c); in Figure 5(d), the storage cost of the CEABKS-MA system in *Encrypt* is slightly higher than that of the HP-CPABKS system; due to $|U_j| < |U|$, the ciphertext storage cost of the CEABKS-MA system is still limited.

7. Conclusion

In this paper, we propose a secure and efficient CEABKS-MA system to support IoV data sharing. The cloud-edge-end collaborative search architecture is designed to meet the real-time search requirements of users and alleviate the severe computation and storage overload problem in the cloud. The multiauthority structure is designed to effectively avoid single-point performance bottlenecks. In addition, the proposed system implements fine-grained keyword search for specified vehicle attributes and extends lightweight encryption and decryption to support attribute updates. Then, this paper demonstrates that the CEABKS-MA system can achieve IND-CPA and IND-CKA security. Experimental simulations prove that the proposed system can effectively reduce computation and storage costs. Since the search query of users is diverse and personalized, on the basis of protecting user privacy, we will dig deeper into users' search intentions and provide users with more intelligent search results.

Data Availability

This article is based on the PBC cryptography library for verification; the real data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61901071, 61871062, 61771082, U20A20157), the General Project of Natural Science Foundation of Chongqing (cstc2019jcyj-msxmX0303), the Science and Natural Science Foundation of Chongqing, China (cstc2020jcyj-zdxmX0024), the University Innovation Research Group of Chongqing (CXQT20017), and the Program for Innovation Team Building at Institutions of Higher Education in Chongqing (CXTDX201601020).

References

[1] M. B. Mollah, J. Zhao, D. Niyato et al., "Blockchain for the Internet of Vehicles towards intelligent transportation systems: a survey," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4157–4185, 2021.

[2] F. Tang, B. Mao, N. Kato, and G. Gui, "Comprehensive survey on machine learning in vehicular network: technology, applications and challenges," *IEEE Communications Surveys and Tutorials*, vol. 23, no. 3, pp. 2027–2057, 2021.

[3] B. Cao, Z. Sun, J. Zhang, and Y. Gu, "Resource allocation in 5G IoV architecture based on SDN and fog-cloud computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3832–3840, 2021.

[4] Y. Wang, Y. Tian, X. Hei, L. Zhu, and W. Ji, "A novel IoV block-streaming service awareness and trusted verification scheme in 6G," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 5197–5210, 2021.

[5] Z. Yang, R. Wang, D. Wu, B. Yang, and P. Zhang, "Blockchain-enabled trust management model for the Internet of Vehicles," *IEEE Internet of Things Journal*, p. 1, 2021.

[6] L. Zhu, H. Liang, H. Wang, B. Ning, and T. Tang, "Joint security and train control design in blockchain empowered CBTC system," *IEEE Internet of Things Journal*, p. 1, 2021.

[7] L. Zhu, Y. Li, F. R. Yu, B. Ning, T. Tang, and X. Wang, "Cross-layer defense methods for jamming-resistant CBTC systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 7266–7278, 2021.

[8] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceeding 2000 IEEE Symposium on Security and Privacy*, pp. 44–55, Berkeley, CA, USA, May 2000.

[9] Y. Yu, L. Guo, S. Liu, J. Zheng, and H. Wang, "Privacy Protection Scheme Based on CP-ABE in Crowdsourcing-IoT for

Smart Ocean," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10061–10071, 2020.

[10] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proceedings of the 14th ACM conference on Computer and communications security*, vol. 2007, pp. 195–203, Virginia, Alexandria, USA, October 2007.

[11] H. Cui, R. H. Deng, G. Wu, and J. Lai, "An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures," in *International Conference on Provable Security*, Springer, Cham, 2016.

[12] X. Mao, J. Lai, Q. Mei, K. Chen, and J. Weng, "Generic and efficient constructions of attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 5, pp. 533–546, 2016.

[13] Q. Zheng, S. Xu, and G. Ateniese, "VABKS: verifiable attribute-based keyword search over outsourced encrypted data," in *IEEE INFOCOM 2014-IEEE conference on computer communications*, pp. 522–530, Toronto, ON, Canada, May 2014.

[14] W. Sun, S. Yu, W. Lou, Y. T. Hou, and H. Li, "Protecting your right: verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 4, pp. 1187–1198, 2016.

[15] T. Tang, L. Li, X. Wu et al., "TSA-SCC: text semantic-aware screen content coding with ultra low bitrate," *IEEE Transactions on Image Processing*, vol. 31, pp. 2463–2477, 2022.

[16] X. Hou, Z. Ren, J. Wang et al., "Reliable computation offloading for edge-computing-enabled software-defined IoV," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7097–7111, 2020.

[17] D. Wu, X. Han, Z. Yang, and R. Wang, "Exploiting transfer learning for emotion recognition under cloud-edge-client collaborations," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 479–490, 2021.

[18] D. Wu, J. Yan, H. Wang, and R. Wang, "User-centric edge sharing mechanism in software-defined ultra-dense networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 7, pp. 1531–1541, 2020.

[19] P. N. Zhang, X. F. Li, D. P. Wu, and R. Y. Wang, "Edge-cloud collaborative entity state data caching strategy toward networking search service in CPSs," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 10, pp. 6906–6915, 2021.

[20] Y. Li, L. Zhu, H. Wang, F. R. Yu, and S. Liu, "A cross-layer defense scheme for edge intelligence-enabled CBTC systems against MitM attacks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2286–2298, 2021.

[21] P. Zhang, Y. Chui, H. Liu, Z. Yang, D. Wu, and R. Wang, "Efficient and privacy-preserving search over edge-cloud collaborative entity in IoT," *IEEE Internet of Things Journal*, p. 1, 2021.

[22] P. V. Liesdonk, S. Sedghi, J. Doumen, P. Hartel, and W. Jonker, "Computationally efficient searchable symmetric encryption," in *Workshop on Secure Data Management*, vol. 6358, pp. 87–100, Springer, Berlin, Heidelberg, 2010.

[23] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," *Journal of Computer Security*, vol. 19, no. 5, pp. 895–934, 2011.

[24] S. Kamara, C. Papamanthou, and T. Roeder, "Dynamic searchable symmetric encryption," in *Proceedings of the 2012 ACM conference on Computer and communications security*, pp. 965–976, North Carolina, Raleigh, USA, October 2012.

[25] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Publickey encryption with keyword search," in *International conference on the theory and applications of cryptographic techniques*, pp. 506–522, Springer, Berlin, Heidelberg, 2004.

[26] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 457–473, Springer, Berlin, Heidelberg, 2005.

[27] 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*, pp. 89–98, Los Angeles , CA , USA, October 2006.

[28] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE symposium on security and privacy*, pp. 321–334, Berkeley, CA, USA, May 2007.

[29] S. Qiu, J. Liu, Y. Shi, and R. Zhang, "Hidden policy ciphertext-policy attribute-based encryption with keyword search against keyword guessing attack," *Science China Information Sciences*, vol. 60, no. 5, 2017.

[30] Y. Miao, J. Ma, Q. Jiang, X. Li, and A. K. Sangaiah, "Verifiable keyword search over encrypted cloud data in smart city," *Computers & Electrical Engineering*, vol. 65, pp. 90–101, 2018.

[31] K. Zhang, J. Long, X. Wang, H.-N. Dai, K. Liang, and M. Imran, "Lightweight searchable encryption protocol for industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4248–4259, 2021.

[32] Q. Wu, J. Domingo-Ferrer, and U. González-Nicolás, "Balanced trustworthiness, safety, and privacy in vehicle-to-vehicle communications," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 2, pp. 559–573, 2010.

[33] R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, and N. Kumar, "P2SF-IoV: a privacy-preservation-based secured framework for Internet of Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2021.

[34] L. Zhou, L. Yu, S. Du, H. Zhu, and C. Chen, "Achieving differentially private location privacy in edge-assistant connected vehicles," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4472–4481, 2019.

[35] J. Kang, R. Yu, X. Huang, and Y. Zhang, "Privacy-preserved pseudonym scheme for fog computing supported Internet of Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2627–2637, 2018.

[36] Y. Wu, L. P. Qian, H. Mao et al., "Secrecy-driven resource management for vehicular computation offloading networks," *IEEE Network*, vol. 32, no. 3, pp. 84–91, 2018.

[37] W. Chen, Y. Chen, X. Chen, and Z. Zheng, "Toward secure data sharing for the IoV: a quality-driven incentive mechanism with on-chain and off-chain guarantees," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1625–1640, 2020.

[38] J. Cui, F. Ouyang, Z. Ying, L. Wei, and H. Zhong, "Secure and efficient data sharing among vehicles based on consortium blockchain," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2021.

[39] H. Wang, K. Fan, K. Zhang, Z. Wang, H. Li, and Y. Yang, "Encrypted data retrieval and sharing scheme in space-air-

ground integrated vehicular networks," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 5957–5970, 2021.

[40] J. Zhang, T. Li, M. S. Obaidat, C. Lin, and J. Ma, "Enabling efficient data sharing with auditable user revocation for IoV systems," *IEEE Systems Journal*, vol. 16, no. 1, pp. 1355–1366, 2022.

[41] C. Feng, K. Yu, M. Aloqaily, M. Alazab, Z. Lv, and S. Mumtaz, "Attribute-based encryption with parallel outsourced decryption for edge intelligent IoV," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13784–13795, 2020.

[42] Y. Chen, X. Liao, and K. Wong, "Chosen plaintext attack on a cryptosystem with discretized skew tent map," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 7, pp. 527–529, 2006.

[43] H. Wang, X. Dong, and Z. Cao, "Multi-value-independent ciphertext-policy attribute based encryption with fast keyword search," *IEEE Transactions on Services Computing*, vol. 13, no. 6, pp. 1142–1151, 2020.

[44] B. Waters, "Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization," in *International workshop on public key cryptography*, pp. 53–70, Springer, Berlin, Heidelberg, 2011.