Toward Achieving Fine-Grained Access Control of Data in Connected and Autonomous Vehicles

Jie Cui, *Member, IEEE*, Xuelian Chen, Jing Zhang, *Graduate Student Member, IEEE*, Qingyang Zhang, *Member, IEEE*, and Hong Zhong, *Member, IEEE*

***Abstract*—A connected and autonomous vehicle (CAV) is often fitted with a large number of onboard sensors and applications to support autonomous driving functions. Based on the current research, little work on applications' access to in-vehicle data has been done. Furthermore, most existing autonomous driving operating systems lack authentication and encryption units. As such, applications can excessively obtain confidential information, such as vehicle location and owner preferences and even upload it to the cloud, threatening the security of the vehicle and the privacy of the owner. In this study, we propose a fine-grained access control scheme to restrict applications' access to data in CAVs (FGAC-inCAVs). First, we present a system model composed of the following elements: a trusted third party (TTP), which is a fully trusted authority; perception components like sensors, which can capture the road information (pictures, videos, etc.); and multiple applications. Then, a fast attribute-based encryption (ABE) is presented, and security analysis also shows it is secure against selective and chosen-plaintext attacks. Furthermore, we propose a key update scheme based on the Chinese remainder theorem (CRT). Finally, the theoretical analysis and simulation experiments demonstrate its feasibility and efficiency.**

***Index Terms*—Access control, attribute-based encryption (ABE), connected and autonomous vehicles (CAVs), security and privacy.**

I. Introduction

THE INTELLIGENT autonomous vehicle is the ultimate development direction of future intelligent vehicles. The connected and autonomous vehicle (CAV) is a system that integrates environmental perception, planning, decision making, multilevel driving assistance, and other functions. We can consider CAV to be a sophisticated mobile computer, which has a large number of onboard sensors and a variety of applications or services to support autonomous driving functions, such as positioning, navigation, steering, braking, etc. [1]–[3]. Moreover, people often want to install entertainment applications to make driving more enjoyable. Currently, most existing autonomous driving vehicles utilize the robot operating system (ROS) as a communication middleware that facilitates communications between different service parts [4]. The controller area network (CAN) protocol is also a popular standard for the in-vehicle networks [5], [6]. However, these networks suffer from a grave security problem: there is no authentication unit for message passing [7]. Hence, an attacker can target a sensor, a control unit, or a computing unit maliciously and even assume control of the vehicle, leading to serious traffic accidents [8]. With the multitude of communication interfaces, it is very difficult, or even impossible, to reliably control entry points into the vehicle or shield the vehicular network with firewalls.

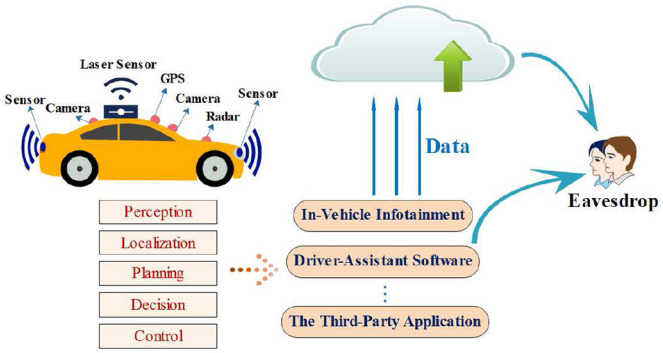


Fig. 1. Model of CAVs' internal applications leaking information.

Safety is the most important requirement for CAVs because there can be no human intervention by drivers [9]. As mentioned above, to ensure the normal operation of the vehicle, dozens of applications are assembled, generating a mass of data in real time [10]. As shown in Fig. 1, large applications have access to sensitive data and resources on the CAV, risking access to the vehicle's confidential information by unrelated applications or even their upload to remote cloud servers [11], [12]. In this way, attackers can use specific applications to steal the core data of the vehicle and then leak the private user information. An example of this would be the unauthorized access, by a vehicle-mounted music application, to onboard GPS information, compromising the vehicle's location information and exposing the CAV to security and privacy risks; such access may even cause traffic accidents, threatening both personal and public safety.

The aforementioned threat arises from the fact that CAV users cannot control the access capacities of CAV applications to sensitive resources. To meet the increasing demand for reliability, a CAV should have the ability to better control

Manuscript received September 10, 2020; revised October 24, 2020; accepted November 23, 2020. Date of publication December 2, 2020; date of current version May 7, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61872001, Grant 62011530046, and Grant U1936220; and in part by the Open Fund of Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University under Grant ESSCKF2018-03. *(Corresponding author: Hong Zhong.)*

The authors are with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui Engineering Laboratory of IoT Security Technologies, and Institute of Physical Science and Information Technology, Anhui University, Hefei 230039, China (e-mail: zhongh@ahu.edu.cn).

Digital Object Identifier 10.1109/JIOT.2020.3041860

applications' access to resources within the vehicle, avoiding situations where applications have excessive permission [13]. For instance, in-vehicle video can provide road relaxation services for passengers, but this service does not need to access sensor perception data necessary for navigation applications, such as road obstacle picture or traffic environment video to name a few. Thus, we can further enhance the privacy and security of the vehicle by restricting unrelated applications' access to internal data. Therefore, it is crucial for CAVs to properly manage applications' access to in-vehicle data and providing a fine-grained data access control scheme for CAVs is truly essential to maintain their security.

Previous research on CAVs has mainly focused on path planning and road navigation. Some works have also studied the perception of autonomous vehicles of the road environment, scene modeling, and vehicle obstacle avoidance. Unfortunately, however, very little research has been done on the security of CAVs [7]. Video or other data can be accessed by multiple applications simultaneously and each application can access many types of data according to its needs. Adopting an access-control algorithm suitable for CAVs to manage the data access of applications reasonably can effectively improve their security and privacy. Despite its benefits, the access control scheme for CAVs is challenging as it should satisfy the following requirements we have identified.

1) *R1:* The scheme should flexibly manage the permission of the applications to in-vehicle data and deal with complex access control requirements.
2) *R2:* None of the applications should be able to bypass the access policy restrictions applied to the vehicle. Applications can only access the relevant vehicle resources within the scope of their authority.
3) *R3:* The additional cost brought on by the security alorithm should not be too large as it will negatively impact the performance and affect the normal function of the vehicle.

In this study, we provide a privacy-preserving fine-grained access control scheme for CAVs that effectively manages applications' access to vehicle private data. Through this scheme, applications can only obtain data according to their access policies. The novelty of our proposal lies in presenting a fast key-policy attribute-based encryption (KP-ABE) algorithm and applying it to autonomous driving for the first time, achieving the isolation and flexible distribution of in-vehicle data. Our main idea is to use the advanced encryption standard (AES) and attribute-based encryption (ABE) to encrypt the vehicle perception data (pictures, videos, etc.) and allow only the applications that satisfy the access policy to obtain their relevant data. Herein, we emphasize that we only deal with the real-time data stream, not the massive amount of data that is previously stored in the CAV, as it has little significance for the encryption of historical perception data with the change of vehicle location. Furthermore, the computing capacity of the vehicle is limited and encrypting only the real-time data stream to achieve privacy protection can save costs. The main contributions of this article are as follows.

1) To the best of our knowledge, this is the first access control scheme for CAVs. With the proposed scheme, applications will be unable to access unauthorized data beyond their scope of service, improving the security and privacy of CAVs. Furthermore, theoretical analysis and simulation experiments show its validity and feasibility.
2) We propose a fast KP-ABE algorithm because it reduces the encryption/decryption requirements to one pairing/two pairings; its implementation also shows a better performance compared to other security schemes.
3) The proposed scheme allows users to personalize access policies so that each CAV user can define preferences according to their needs. In addition, we present a key update strategy based on the Chinese remainder theorem (CRT).

The remainder of this article is organized into eight sections. Section II introduces some related work. Section III describes preliminary knowledge and technical background relevant to this study. Afterward, we present the system model and the attack model in Section IV. Then, we give the detailed scheme in Section V and the specific security analysis of the proposed scheme in Section VI. Next, the cost of computation and communication is discussed in Section VII. The implementation in Section VIII shows the feasibility of the scheme. Finally, we carry on the conclusion in Section IX.

II. Related Work

In this section, we introduce some previous works from autonomous vehicles, ABE, and its application in pub/sub system.

Jo *et al.* [14], [15] presented the autonomous distributed driving system, a development process, and a system platform for the heterogeneous computing system. At the same time, Jo *et al.* expounded the core algorithm of driverless vehicles and the feasibility and effectiveness of their schemes by combining case studies. In recent years, many scholars have done a lot of research on vehicle path planning, 3-D modeling, automatic obstacle avoidance, and other behaviors [16]–[20]. However, there is little research on the security of internal resource access of autonomous vehicles.

Sahai and Waters [21] first introduced the notion of ABE, then two types of attribute encryption schemes: KP-ABE and ciphertext-policy ABE (CP-ABE) were formally and systematically defined by Goyal *et al.* [22] and Bethencourt *et al.* [23]. In a KP-ABE system, the ciphertext is labeled with a set of specific attributes and the decryption key is associated with an access structure, only the party whose attributes match the access policy can acquire the key to decrypt the message. Ostrovsky *et al.* [24] further extended and proposed an ABE scheme with nonmonotonic access structures that can handle any access formula, these were applied to achieve practical broadcast encryption by Lewko *et al.* [25] and Boneh *et al.* [26]. Since then, more scholars have been committed to fast encryption/decryption, outsourced encryption and decryption, and unbounded ABE [27], [28].

Thanks to the characteristics of ABE, it has been widely applied to numerous fields. A popular instance is that to

protect sensitive messages, several works have been done to apply ABE to publish and subscribe (pub/sub) system, which defines a one-to-many dependency that allows multiple subscriber objects to listen on a topic object at the same time [29]. Pal *et al.* [30] presented P3S, a publish–subscribe middleware mainly adopting CP-ABE and PBE, to protect the privacy of the subscriber interest and confidentiality of published content. Thatmann *et al.* [31] used a group controller and ABE to encrypt data on update procedures to protect communication in Internet of Things (IoT). Then, Yang *et al.* [32] introduced a privacy protection scheme for cloud platforms so that the publishers can manage the access to messages and the privacy of the subscriber can also be protected. Furthermore, the scheme in [33] can resist collusion attacks between untrusted brokers and malicious publishers or subscribers.

III. Technical Background

In this section, we briefly cover the preliminary knowledge used in the proposed scheme.

A. Key-Policy Attribute-Based Encryption

In the KP-ABE scheme, the ciphertext is marked by an attribute set published by the authority while the key is generated according to an access structure which can be defined by users. The data owner can use KP-ABE to encrypt the information and label the ciphertext with attribute sets. Only the users satisfying the access policy can gain the private key to decrypt the corresponding ciphertext. A KP-ABE scheme consists of the following polynomial-time algorithms.

Set-Up $\lambda \rightarrow$ *(PK,MK):* This randomized algorithm takes a security parameter 1^{λ} as input, and outputs the public parameter *PK* as well as a master key *MK*.

Encryption (PT,γ*,PK)* $\rightarrow$ *CT:* The encryption algorithm is a randomized algorithm, it inputs a message PT in plaintext, a set of attributes γ used in this step, and the public parameter *PK* generated in Set-up, then it outputs the ciphertext *CT* associated with the attribute set.

KeyGen (AS,MK) $\rightarrow$ *sk:* This randomized algorithm is to generate the private key for decryption. It generates a private key *sk* based on an access tree *AS* and the master key *MK* [34].

Decryption (CT,sk) $\rightarrow$ *PT:* This is a deterministic algorithm which inputs the ciphertext *CT* associated with the attributes γ, a private key *sk* associated with the access structure *AS*, then it can output the message PT if the attribute set γ satisfies *AS*.

B. Linear Secret-Sharing Schemes

We first introduce a definition as follows.

Definition: A scheme Π that includes several parties P_a is called a linear secret-sharing scheme (LSSS) (over Z_p) if it satisfies the following conditions.

1) Each shares of the secret can form a vector in the Z_p.
2) There is a share-generating matrix M with l rows and $n+1$ columns. The map δ which is responsible for mapping the ith$(i = 1, \ldots, l)$ row of M to a party x'_i ($x'_i \in P_a$). Then, the column vector is $v = (se, c_1, c_2, \ldots, c_n)$, where $se \in Z_p$ is the secret to be shared, and $c_1, \ldots, c_n$ are random numbers chosen from Z_p.

Furthermore, we define a submatrix M_k according to Π, which includes k shares of the secret s.

As defined above, the linear reconstruction property used in LSSS can be described as follows. Suppose that X is an LSSS. *AS* is an access structure constructed according to the policies and S_{attr} ($S_{\text{attr}} \in AS$) is a set of authorized attributes. $P_a \subset \{1, 2, \ldots, n\}$ can be expressed as $P_a = \{i : x_i \in S_{\text{attr}}\}$. If $\{\lambda_i\}$ are valid shares of secret *se*, then we can use a set of constants $\{w_i \in Z_p\}_{i \in P_a}$ to compute the secret, because there exists an equation as $\sum_{i \in P_a} w_i \lambda_i = se$.

C. Decisional Bilinear Diffie–Hellman Assumption

Let G_1, G_2 be two multiplicative cyclic groups of prime order p, it then picks a random generator $g \in G_1$. In addition, let $e : G_1 \times G_1 \rightarrow G_2$ denote a bilinear map. We say that the bilinear map $e : G_1 \times G_1 \rightarrow G_2$ is efficiently computable and symmetric, so there exists an equation $e(g^j, g^k) = e(g, g)^{jk} = e(g^j, g^k)$. We now state an assumption as follows.

The decisional bilinear Diffie–Hellman (BDH) assumption is that the tuple $(g, H = g^h, I = g^i, J = g^j, e(g, g)^{hij})$ and the tuple $(g, H = g^h, I = g^i, J = g^j, e(g, g)^z)$ cannot be distinguished by a probabilistic polynomial-time (PPT) algorithm Q with more than a negligible advantage [35]. Herein, the advantage of Q is defined as

$$\left|\Pr\left[Q\left(H, I, J, e(g, g)^{hij} = 0\right\right] - \left[Q\left(H, I, J, e(g, g)^z\right) = 0\right]\right|$$

where h, i, j, z in Z_p, the generator g in G_1 is randomly chosen, and the random bits are consumed by Q. So the advantage has probability.

D. Chinese Remainder Theorem

Let $j_1, j_2, j_3, \ldots, j_n$ be n pairwise relative prime positive integers, and let $k_1, k_2, \ldots, k_n$ be n arbitrary integers. Then, CRT states that the pair of congruences

$$\begin{aligned} X_{\text{CRT}} &\equiv k_1 (\text{mod } j_1) \\ X_{\text{CRT}} &\equiv k_2 (\text{mod } j_2) \\ &\vdots \\ X_{\text{CRT}} &\equiv k_n (\text{mod } j_n) \end{aligned}$$

has a unique solution modulo $j_1, j_2, j_3, \ldots, j_n$. To compute the unique solution, suppose $J = j_1, j_2, j_3, \ldots, j_n$, we can compute the unique solution X_{CRT} as shown: $X_{\text{CRT}} = \sum_{i=1}^{n} j_i J_i J'_i (\text{mod } J)$, where $J_i = J/j_i$ and J'_i is the multiplicative inverse of $(J_i \text{ mod } j_i)$, i.e., $J_i J'_i \equiv 1 (\text{mod } n)$.

IV. System Model and Attack Model

In this section, we first present the system model of our scheme. There are mainly three entities: 1) trusted third party (TTP); 2) applications; and 3) perception components. Then, we briefly introduce the attack model.

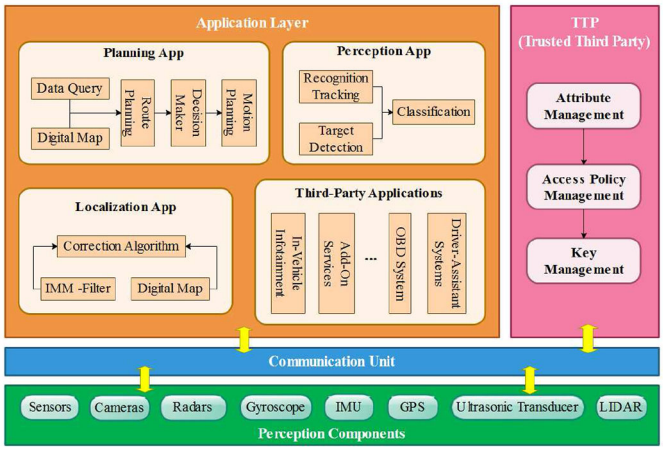


Fig. 2. Distributed system of CAV.

A. System Model

As shown in Fig. 2, the proposed privacy-preserving data access control scheme involving the following entities.

Trusted Third Party: TTP is a fully trusted authority and responsible for access policy, key, attributes management and key, and attributes distribution.

Applications: It includes many service modules which need to acquire environmental information, road conditions, and so on to maintain the normal operation of vehicles. These applications will make every effort to obtain unauthorized resources, so there is a risk of privacy leakage. In our scheme, users can customize the access rights of the applications, and only the authorized application can obtain the key of relevant content for decryption.

Perception Components: It consists of a lot of perception subassembly (such as sensors, radars, GPS, etc.) to collect video image information of the road, vehicle location information, and so on, it also conducts information fusion processing to support the upper application after capturing environmental information. These perception components can use AES to encrypt data in our scheme.

Communication Unit: It provides a connection for sharing information between application softwares, sensors, and actuators.

B. Attack Model

In our scheme, we assume the TTP is a fully trusted authority and the communication channels between entities are secure. Applications are not malicious but can be malicious adversary used to try to access resources beyond their privileges and collude with other unauthorized applications.

Our encryption scheme can resist selective attack and chosen-plaintext attacks by most of the existing literature on ABE. Below is the formal definition.

Definition 1: The ABE scheme is secure under the selective model and chosen-plaintext attack model, if the probability that any adversary $\widetilde{A}$ win the game defined below is no more than a negligible advantage in any polynomial time.

Game: We build a game played between a challenger, an adversary $\widetilde{A}$, and a simulator $\widetilde{B}$ to prove the security of the main construction in the proposed scheme. This game will run in the selective-set model based on the assumption that the decisional BDH problem is hard to solve. The game is defined in these steps.

1) *Init:* First, the adversary $\widetilde{A}$ declares the set of attributes γ to be challenged upon and submits the challenge ciphertext policy to the challenger.
2) *Setup:* The challenger inputs a security parameter 1^{λ} and runs the Set-up algorithm of ABE, then it passes the master key MK and the public parameters PK to the adversary $\widetilde{A}$.
3) *Phase 1:* The adversary $\widetilde{A}$ is allowed to adaptively make a polynomial number of queries for many private keys, where attributes γ may not satisfy the access structures. Then, the simulator $\widetilde{B}$ runs its own key generation oracle and the query result will be returned to the adversary $\widetilde{A}$.
4) *Challenge:* The adversary $\widetilde{A}$ submits two messages of equal length M_0 and M_1. The challenger randomly encrypts the message M_c ($c \in \{0, 1\}$) under the attribute set γ. Then, the ciphertext C computed by M_c is given to the adversary.
5) *Phase 2:* The adversary $\widetilde{A}$ makes queries as in Phase 1 does with some restrictions.
6) *Guess:* The adversary $\widetilde{A}$ outputs a guess c' of c. In the selective-set model, the adversary should submit the challenge policy to the challenger before the Setup phase. But there is no need to add an Init phase when we prove the proposed scheme is CPA-secure.

V. Proposed Scheme

This section details the proposed scheme, that mainly includes five phases: 1) system initialization; 2) key generation; 3) data encryption; 4) data decryption; and 5) key update.

The main purpose of the proposed scheme is to build a resource fine-grained access-control scheme for applications in CAVs to improve privacy and security [36], [37] as demonstrated in Fig. 3. The proposed scheme designs a fast KP-ABE algorithm based on LSSS, in which encryption requires only one bilinear pairing operation and decryption requires two pairings. We take advantage of AES to encrypt the real-time perception data inside an autonomous vehicle, then ABE encrypts the key of AES to achieve fine-grained access control of data. The TTP is in charge of the management of attributes, access policies, and the distribution of keys to the qualified applications to decrypt the corresponding ciphertext. What is more, we also analyze the update of the key. Finally, our focus is to create a more secure fine-grained access-control algorithm that allows users to personalize and flexibly limit the applications' access to resources within a CAV [38]. The definitions and related notations are shown in Table I and the process of communication in the proposed scheme is presented in Fig. 5.

A. System Initialization

Let $g \in G_1$ be a random generator to build a bilinear group G_1 of prime order p, the size of the groups will be determined by the system security parameter 1^n. In addition, we denote a bilinear map $e : G_1 \times G_1 \rightarrow G_2$. What is more, we describe an ABE algorithm with universe U. Herein, the attribute strings

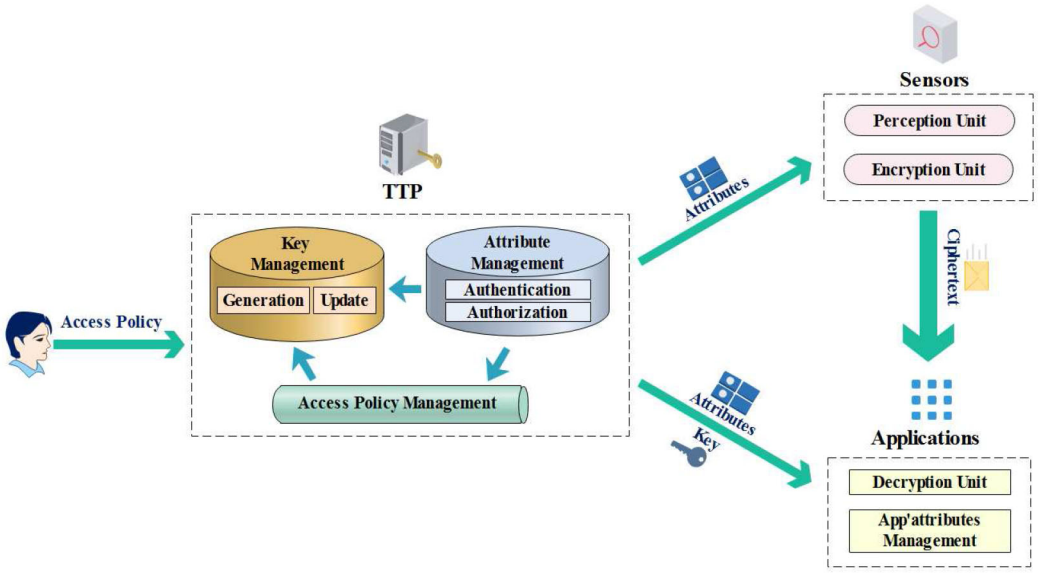


Fig. 3. Framework of the proposed access-control system.

TABLE I
DEFINITIONS OF NOTATIONS

Notations	Definitions
Z_q	Group $\{0, ..., q-1\}$ under multiplication module q (a prime).
a, b, r_i	Random number from Z_p.
e	Bilinear map $e : G_1 \times G_1 \rightarrow G_2$.
U	Universe of attributes in the proposed scheme.
n	The number of attributes of each authorized application.
α	The secret to be shared, $\alpha = a * b$.
H_1, H_2	Hash function.
PK	Public parameter in our ABE scheme.
MK	Master key in our ABE scheme.
M	Data in plaintext.
Ma	Share-generating matrix of LSSS Π.
x_i	the attribute underlying i'th row of Ma.
λ_i	Shares of secret α.
AS	Access structure generated by TTP.
ID_{App}	Identifier of application.
d	The number of attributes with ciphertext C.
K_i	Decryption key materials.
SK	Decryption key set.
γ	Attribute set with ciphertext.
gk	Group key in Key Update.
pk_i	Private key in Key Update.

can be transformed into a series of integers $\{1, \ldots, U\}$ in $Z_p{}^*$ by a collision resistant hash function $H : \{0, 1\}^* \rightarrow Z_p{}^*$.

1) Parameters Set Up: TTP first runs the Setup algorithm to construct a bilinear map e. Then, it chooses $a, b \in Z_p$ randomly, and lets $g_1 = g^a$, $g_2 = g^b$, the secret α is set as $\alpha = a \cdot b$. TTP chooses two hash functions $H_1 : \{0, 1\}^* \rightarrow Z_p$, $H_2 : \{0, 1\}^* \rightarrow G_1$, as well. Finally, it generates a system master key MK, and public parameters PK as

$$MK = \alpha \tag{1}$$

$$PK = (g, g_1, g_2, H_1(\cdot), H_2(\cdot), e(g, g)^{\alpha}). \tag{2}$$

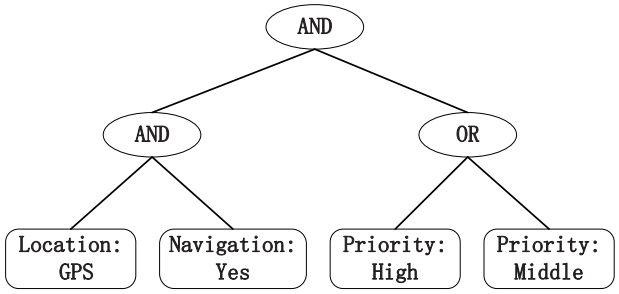


Fig. 4. Example of the access structure.

2) Access Policy and Attributes Generation (Access Structure): Let following be a set of parties: $\{P_1, \ldots, P_n\}$, an access structure (resp., monotone access structure) is a collection (resp., monotone collection) AS of nonempty subsets of $\{P_1, \ldots, P_n\}$, i.e., $AS \subset 2^{\{P_1, \ldots, P_n\}}$.The sets in AS are called the authorized sets, and the sets not in AS are called the unauthorized sets. In our context, the role of the parties is taken by the attributes. Thus, the access structure AS will contain the authorized sets of attributes.

Users can personalize a set of access control policies for fine-grained management of applications' access to system resources, further enhancing system security and privacy. So TTP needs to generate a set of attributes U and manage these authorized attribute sets.

Furthermore, as shown in Fig. 4, it generates a set of access structures AS represented as a collection of trees according to the access policy, which is to be used in the key generation for the next step. Each leaf of the access tree is labeled with a pair (*attribute* : *value*) and the nonleaf nodes represent threshold gates, e.g., "AND" and "OR." Furthermore, the access structure AS can be convert to (Ma, Γ), where Ma is an $l \times (n+1)$ share-generating matrix for LSSS Π and Γ is a function that maps from the rows of the matrix to corresponding attributes. Remarkably, each row of Ma is associated with an attribute,

and we use x_i to represent the attribute underlying ith row of Ma.

3) Application Authentication: Every application should be authenticated by TTP and identified as a tuple (ID_{App}, θ), where ID_{App} is an identifier made up of a unique serial number, and θ is a collection of tags which can be written as $\theta = \{\theta_1, \theta_2, \ldots, \theta_m\}$. Notably, we let each tag θ be defined as (*attribute*: *value*). For example, an advanced driver-assistant application may be expressed as [$ID_{\text{App}} = 0708$, θ = (*Location*: *GPS*, *Recognition*: *Yes*)], here "0708" is the application's identifier assigned by TTP, "Location: GPS, Recognition: Yes" means the corresponding value of the attribute "Location" is "GPS" and the attribute "Recognition" has the value "Yes."

B. Key Generation and Distribution

In this step, TTP will generate a decryption key according to the access structure AS, then distribute these keys to the authorized applications whose tag attributes satisfy the corresponding access structure (i.e., $AS = 1$). The processes are explained in detail as follows.

1) Key Generation: TTP generates the decryption key according to the access policy. If a ciphertext's label attributes satisfy the access structure AS ($AS = 1$), then the key can decrypt this ciphertext. First, TTP converts the access policy to an access structure AS; then, we set α as the secret to be shared and apply LSSS II to obtain shares $\{\lambda_i\}$. The ith($i = 1, \ldots, l$) row of Ma is associated with attribute $x_i \in \gamma$. For each i, we also randomly choose a value $r_i \in Z_p$. Then, the TTP generates the decryption key materials as follows:

$$K_i = \left(K_i^{(1)} = g^{\lambda_i} \cdot H_2(x)^{H_1(x)\cdot r_i}, K_i^{(2)} = g^{H_1(x)\cdot r_i}\right). \quad (3)$$

The key $SK = \{K_i^{(1)}, K_i^{(2)}\}$.

2) Key Distribution: TTP will distribute private key SK to those applications which possess an attribute set S that satisfies the access structure, i.e., $AS(S) = 1$. This ensures that only applications that meet user-defined access policies can obtain the key to decrypt the relevant data.

C. Data Encryption

The sensor module receives a collection of authorized attributes, next it encrypts the perception data in real time with AES, then encrypts the key of AES under a set of d attributes $\gamma \in U$. (We assume that the sensor only processes the real-time data stream, main reason is that the historical data stored in the vehicle is too expensive to encrypt and the encryption is of little significance because the vehicle's data are time sensitive.) The sensor module chooses a value s from Z_p randomly and a random set $\{s_x\}_{x\in\gamma}$, where $s = \sum_{x\in\gamma} s_x$. It outputs the ciphertext with the attribute set as

$$E = \left(\gamma, E^{(1)} = M \cdot e(g_1, g_2)^s, E^{(2)} = g^s \left\{E_x^{(3)} = H_2(x)^{s_x}\right\}_{x\in\gamma}\right). \quad (4)$$

D. Data Decryption

If ciphertext's label does not satisfy AS, then the message cannot be decrypted by application. Otherwise, when $AS(\gamma) = 1$, applications that hold the relevant keys can gain access permission. Suppose γ satisfies AS, we make $\zeta_I = \{i : x_i \in \gamma\}$ be a series of arguments and $\{w_i\}_{i\in\zeta_I} \in Z_p$ be a set of constants such that $\sum_{i\in\zeta_I} w_i \cdot \lambda_i = \alpha$, decryption procedure computes as follows:

$$\begin{aligned} D_i &= \frac{e\left(E^{(2)}, \mathrm{K}_i^{(1)}\right)}{e\left(\prod_{x\in\gamma} E_x^{(3)}, \mathrm{K}_i^{(2)}\right)} \\ &= \frac{e\left(g^s, g^{\lambda_i} \cdot H_2(x)^{H_1(x)r_i}\right)}{e\left(\prod_{x\in\gamma} H_2(x)^{s_x}, g^{r_i H_1(x)}\right)} \\ &= \frac{e\left(g^s, g^{\lambda_i}\right) \cdot e\left(g^s, H_2(x)^{r_i H_1(x)}\right)}{e\left(H_2(x)^{\sum_{x\in\gamma} s_x}, g^{r_i H_1(x)}\right)} \\ &= \frac{e\left(g^s, g^{\lambda_i}\right) \cdot e\left(g^s, H_2(x)^{r_i H_1(x)}\right)}{e\left(H_2(x)^{r_i \cdot H_1(x)}, g^s\right)} \\ &= e(g, g)^{s\cdot\lambda_i}. \end{aligned} \quad (5)$$

Finally, the applications can decrypt the data by computing

$$\frac{E^{(1)}}{\prod_{i\in\zeta_I} D_i^{w_i}} = \frac{M \cdot e(g_1, g_2)^s}{e(g, g)^{s\cdot\left(\sum_{i\in\zeta_I} \lambda_i w_i\right)}} = M. \quad (6)$$

E. Key Update

In this scheme, two key update strategies based on the CRT is presented: 1) membership change-based update and 2) time-based update [39]–[42]. Now, let us delve into the details of key update strategies.

1) Initialization: TTP picks private keys $pk_1, pk_1, \ldots, pk_n$ for each application, where $pk_1, pk_1, \ldots, pk_n$ are pairwise positive integers; and generates an initial group key gk randomly. Then, we build the congruence system as follows (suppose there are n applications for this group):

$$\begin{aligned} X &\equiv k_1 \left(\text{mod } \mathrm{pk}_1\right) \\ X &\equiv k_2 \left(\text{mod } \mathrm{pk}_2\right) \\ &\vdots \\ X &\equiv k_n \left(\text{mod } \mathrm{pk}_n\right). \end{aligned} \quad (7)$$

Here, k_i is computed as $gk \oplus pk_i$, for all $i \in \{1, 2, \ldots, n\}$. Everytime the X being computed out, TTP can simply submit this value of X to all applications that have been authenticated. Obviously, every application can share the group key gk by simply doing one XOR operations and one modulo.

2) Membership Change-Based Update: When the loaded application changes, the corresponding group secret key should also be updated to ensure that new members get the key and that leaving group members can no longer access the relevant resources.

1) *Member Join:* As mentioned earlier, when an application joins a group, it should first be authenticated by TTP and obtain the corresponding key according to the access structure. Then, TTP will pick a private update key pk'

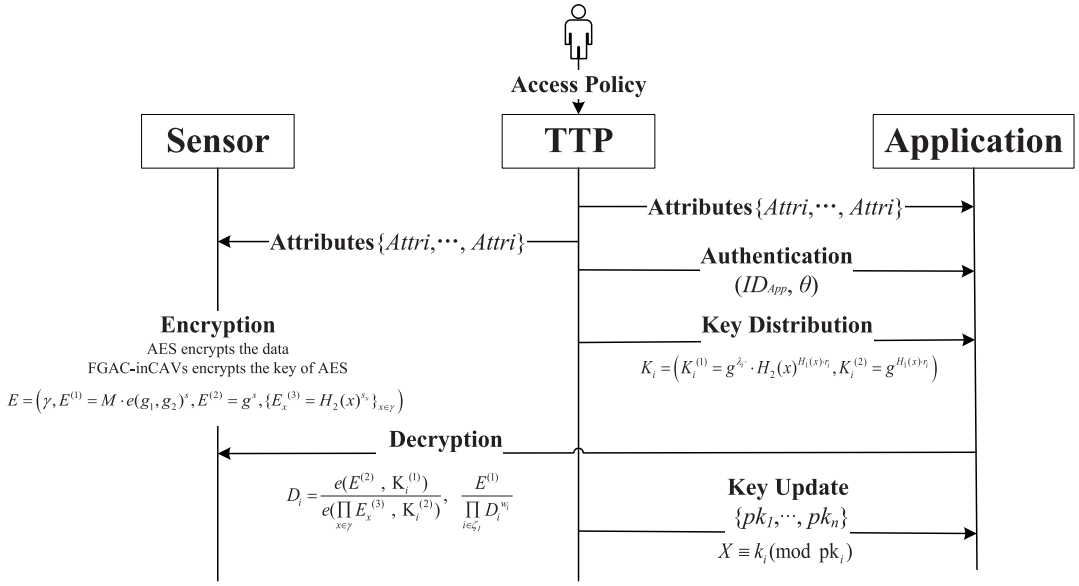


Fig. 5. Process of communication between the sensor, TTP, and application in the proposed access-control scheme.

for the new application (pk' is also a pairwise positive integer) and choose a new group key gk'. We use X' to represent new value of X, it can be updated as follows:

$$\begin{aligned} X' &\equiv k_1 \left(\text{mod } \text{pk}_1\right) \\ &\vdots \\ X' &\equiv k_n \left(\text{mod } \text{pk}_n\right) \\ X' &\equiv k' \left(\text{mod } \text{pk}'\right). \end{aligned} \tag{8}$$

2) *Member Leave:* When members leave, we should update the decryption key of all applications to prevent unauthorized applications from decrypting the corresponding ciphertext using the old key. All we need to do is take the private key pk_i out of our congruent system, choose a new set of keys, then calculate a new X value.

3) Time-Based Update: In addition to the above update strategy, we add a regular update policy to our key update scheme. At certain intervals, TTP will update all the keys to improve the security. In this strategy, TTP only needs to compute the new value of X according to the new group key and broadcasts it to every application. Note that it does not have to recompute the intermediate results such as $M_i M_i', i \in \{1, 2, \ldots, n\}$ every time, these results can be saved to improve the update efficiency.

VI. Security Analysis

In this section, we prove the proposed KP-ABE scheme is safe under the selective model and the CPA model. Then, we analyze the security of the whole access control scheme.

A. Security Proof

We now prove that the security of the proposed ABE against selective attack in the attribute-based selective-set model based on the assumption that the decisional BDH is hard to break.

Theorem 1: If there exists an adversary that can break through our construction in the attribute-based selective-set model, then there can be a simulator to break the decisional BDH assumption with a nonnegligible advantage.

Proof: Suppose there is an adversary $\widetilde{A}$ can break our ABE scheme with advantage ϵ in polynomial time under the selective-set model, and then we can prove that there exists a simulator $\widetilde{B}$ can win the decisional BDH game with a probability of no less than $\epsilon/2$. ■

Let the challenger pick a generator $g \in G_1$ and set an efficient bilinear map $e : G_1 \times G_1 \to G_2$. Then, the challenger flips a fair binary coin φ, and the result cannot be seen by the simulator $\widetilde{B}$. If $\varphi = 0$, the challenger sets the tuple as $(A, B, C, D) = (g^a, g^b, g^c, e(g, g)^{abc})$; otherwise, it makes the tuple as $(A, B, C, D) = (g^a, g^b, g^c, e(g, g)^z)$, where a, b, c, z is randomly chosen from Z_p.

Init: $\widetilde{B}$ runs adversary $\widetilde{A}$, then $\widetilde{A}$ outputs a challenged attribute set γ, which has d elements of Z_p*.

Setup: $\widetilde{B}$ sets the public parameters $g_1 = A = g^a$, $g_2 = B = g^b$ and computes α as $\alpha = a \cdot b$, where a, b are randomly chosen from Z_p. Now, we can write the attribute set as $\gamma = \{x_1, x_2, \ldots, x_d\}$, then $\widetilde{B}$ will program the random oracle $H_1(x), H_2(x)$.

H_1, H_2 Hash Query: When adversary $\widetilde{A}$ makes queries to the oracle on x (x belongs to U). If such a query has been answered by $\widetilde{B}$, $\widetilde{B}$ will merely return the same result. Otherwise, it randomly chooses $z_x, T_x, f_x, t \in Z_p$ and responds as follows:

$$H_1(x) = \begin{cases} T_x, & x \in \gamma \\ z_x T_x, & x \notin \gamma \end{cases} \tag{9}$$

$$H_2(x) = \begin{cases} g^{f_x}, & x \in \gamma \\ g^{f_x} g^t, & x \notin \gamma. \end{cases} \tag{10}$$

Then, the public parameters PK will be assigned as $PK = (g, g_1, g_2, e(g, g)^\alpha, H_1(\cdot), H_2(\cdot))$ and given to adversary $\widetilde{A}$. Observe that all parameters are well distributed due to the

random values of α, z_x, T_x, f_x, t and consistent with the actual distribution.

Phase 1: Adversary $\widetilde{A}$ adaptively makes a set of key queries, no matter wether the challenge attribute sets γ satisfy the access structures or not. Now, suppose $\widetilde{A}$ requests the secret key corresponding to an access tree AS, where $AS(\gamma) = 0$. To successfully generate the decryption key, the simulator $\widetilde{B}$ should to do the following according to the LSSS Π.

$\widetilde{B}$ first parses the access structure AS as (Ma, Γ), where Ma is a $l \times (n+1)$ share-generating matrix for LSSS Π and Γ is a function that maps from the rows of the matrix to corresponding attributes. Remarkably, each row of Ma is associated with an attribute, and we use x_i to represent the attribute underlying the ith row Ma_i. What is more, the column vector can be represented as $\text{col} = (s, c_1, c_2, \ldots, c_n)$, among them s stands for secret to be shared in the secret-sharing scheme ($s = 1$ means that we can reconstruct the secret) and $c_1, c_2, \ldots, c_n$ are randomly chosen from Z_p. In addition, each row of Ma can be treated as a share and we can write the submatrix composed of T shares of s as MT [i.e., for $i \in MT, \Gamma(i) \in \gamma$].

According to the linear secret sharing system Π, if such a column vector $(1, 0, \ldots, 0)$ exists in the span of MT, then these parties associated with MT can reconstruct the secret. To see that this is well defined, consider that since $AS(\gamma) = 0$, then we cannot find the vector $(1, 0, \ldots, 0)$ in the span of M_i. In other words, this vector has no linear dependence of the rows of MT.

To choose a secret, we first define a random vector $\text{vec} = (v_1, v_2, \ldots, v_{n+1}) \in {Z_p}^{n+1}$, which has $n+1$ dimensionality. Thus, such vector that satisfies the proposition only if $Ma \cdot \text{vec} = 0$ and $(1, 0, \ldots, 0) \cdot \text{vec} = v_1 = 1$ then the vector vec is linearly independent of matrix Ma. Next, the simulator $\widetilde{B}$ can generate a specific key used to share the secret with vec. The shares are computed as $\lambda = Ma \cdot \text{vec}$.

Through the above steps, how to distribute shares to entities labeled with attributes has been shown. We also proved these shares have no dependence on a, b. Next, we turn to introducing the step to generate decryption key material.

If $x_i \in \gamma$, λ_i is independent of all secrets, $\widetilde{B}$ randomly chooses a parameter $r_i \in Z_p$, then the key material can be computed as follows:

$$K_i = \left(K_i^{(1)} = g^{\lambda_i \cdot} \cdot H_2(x)^{H_1(x) \cdot r_i}, K_i^{(2)} = g^{H_1(x) \cdot r_i}\right). \quad (11)$$

If $x_i \notin \gamma$, let $g_3 = g^{\lambda_i}$ and randomly pick $w_i' \in Z_p$, set

$$K_i^{(1)} = \left[g_3{}^{-(f_x+t)z_xT_x}\right]^{z_xT_x} g_3\left(g^{f_x}g^t\right)^{z_xT_xw_i'} \quad (12)$$

$$K_i^{(2)} = \left(g^{w_i'} \cdot g_3{}^{-z_xT_x}\right)^{z_xT_x}. \quad (13)$$

Claim 1: The decryption key produced by the simulation above is valid; furthermore, it has identical distribution as the key generated by the actual ABE scheme if the public parameter is same.

Proof: We now prove this claim used the following specific case. ∎

If $x_i \in \gamma$, simulation produces the key material that is same as the key generated by the actual ABE scheme.

Otherwise, $x_i \notin \gamma$, let $r_i = w_i{}' - \lambda_i z_x T_x$, we can compute as follows:

$$\begin{aligned} K_i^{(1)} &= \left[g_3{}^{-(f_x+t)z_xT_x}\right]^{z_xT_x} g_3\left(g^{f_x}g^t\right)^{z_xT_xw_i'} \\ &= g^{\lambda_i}\left(g^{f_x}g^t\right)^{-z_xT_x \cdot z_xT_x \cdot \lambda_i}\left(g^{f_x}g^t\right)^{z_xT_xw_i'} \\ &= g^{\lambda_i}\left(g^{f_x}g^t\right)^{(w_i'-z_xT_x \cdot \lambda_i)\cdot z_xT_x} \\ &= g^{\lambda_i} \cdot H_2(x)^{H_1(x)\cdot r_i} \end{aligned} \quad (14)$$

$$\begin{aligned} K_i^{(2)} &= \left(g^{w_i'} \cdot g_3{}^{-z_xT_x}\right)^{z_xT_x} \\ &= \left(g^{r_i+\lambda_i z_xT_x} \cdot g_3{}^{-z_xT_x}\right)^{z_xT_x} \\ &= g^{r_i z_xT_x} \\ &= g^{r_i \cdot H_1(x)}. \end{aligned} \quad (15)$$

Challenge: Adversary $\widetilde{A}$ will choose two challenge messages of equal length M_0, M_1 and send them to $\widetilde{B}$. Then, $\widetilde{B}$ flips a fair binary coin to get a random result u used to encrypt the message M_u. The ciphertext will be computed as follows:

$$E = \left(\gamma, E^{(1)} = M_u Z, E^{(2)} = C, \left\{E_x{}^{(3)} = C^{f_x}\right\}_{x\in\gamma}\right). \quad (16)$$

If $\varphi = 0$, we have $(A, B, C, D) = (g^a, g^b, g^c, e(g,g)^{abc})$. Therefore, the ciphertext E is a valid random encryption of M_u according to attributes γ.

Otherwise, let $D = e(g,g)^z$; then $E^{(1)}$ can be calculated as $E^{(1)} = M_u D$. Hence, $E^{(1)}$ is an element of group G_2 from $\widetilde{A}$'s view. Meanwhile, the encryption does not leak any information about M_u as well.

Phase 2: The simulator $\widetilde{B}$ just do the same behavior as in Phase 1, except with some restrictions.

Guess: The adversary $\widetilde{A}$ outputs a guess u' of u. If $u' = u$, $\widetilde{B}$ will let $\varphi^* = 0$ to express there was a valid BDH-tuple; else, the simulator will output $\varphi^* = 1$ to show it has been given a random four-tuple.

As mentioned above, if $\varphi = 1$ the adversary cannot gain any information about u. So we get the advantage $Pr[u \neq u' | \varphi = 1] = 1/2$. Since $\varphi^* = 1$ when $u \neq u'$, we finally compute $Pr[\varphi = \varphi^* | \varphi = 1] = 1/2$.

Otherwise, when $\varphi = 0$ the adversary $\widetilde{A}$ only gets a ciphertext. In our assumption, the adversary's advantage in this situation is ϵ. So we have $Pr[u = u | \varphi = 0] = 1/2 + \epsilon$. Since there exists $\varphi^* = 0$ if $u = u'$, finally we can compute out the probability $Pr[\varphi = \varphi^* | \varphi = 0] = 1/2 + \epsilon$.

Consequently, as defined by the adversary advantage function, the probability to break the decisional BDH difficult problem is $1/2(\Pr[\varphi = \varphi^* | \varphi = 1] + \Pr[\varphi = \varphi^* | \varphi = 0]) - 1/2 = 1/2(1/2 + 1/2 + \epsilon) - 1/2 = \epsilon/2$.

B. Security Analysis

1) Message Confidentiality: The proposed scheme compensates for the lack of authentication and encryption units in CAVs. Meanwhile, based on the difficulty of the decisional BDH math problem, the confidentiality of messages is guaranteed. The security of the adopted algorithm has been proved in the previous sections.

TABLE II
COMMUNICATION COST ANALYSIS

	PK	MK	SK	$Ciphertext$
Goyal et al. [22]	$nL_{E_1} + L_{E_2}$	$(n+1)L_{E_1} + L_{Z_p}$	L_{E_1}	$dL_{E_1} + L_{E_2}$
Bethencourt et al. [23]	$3L_{E_1} + L_{E_2}$	$L_{E_1} + L_{Z_p}$	$(2n + 1)L_{E_1}$	$(2d + 1)L_{E_1} + L_{E_2}$
Sahai et al. [24]	$(2n + 2)L_{E_1}$	L_{Z_p}	$5L_{E_1}$	$(2d + 1)L_{E_1} + L_{E_2}$
Hohenberger et al. [27]	$L_{E_2} + (n+1)L_{Z_p}$	$L_{E_2} + (n + 2)L_{Z_p}$	$(\|\Gamma\| + 1)L_{E_1}$	$(d+1)L_{E_1} + L_{E_2}$
Our scheme	$3L_{E_1} + L_{E_2} + L_H$	L_{Z_p}	$2L_{E_1}$	$(d+1)L_{E_1} + L_{E_2}$

TABLE III
COMPUTATION COST ANALYSIS

	$Encryption$	$Decryption$
Goyal et al. [22]	$dE_1 + 2E_2 \approx (1.424 + 0.694d)ms$	$de + 2\|Q\|E_2 \approx (5.086d + 1.424\|Q\|)ms$
Bethencourt et al. [23]	$(2d + 1)E_1 + 2E_2 \approx (2.118 + 1.388d)ms$	$2ne + (2\|Q\| + 2)E_2 \approx (10.172n + 1.424\|Q\| + 1.424)ms$
Sahai et al. [24]	$(2d + 1)E_1 + 2E_2 \approx (2.118 + 1.388d)ms$	$2e$ *or* $3e \approx (10.172$ *or* $15.258d)ms$
Hohenberger et al. [27]	$(d + 1)E_1 + 2E_2 \approx (2.118 + 0.694d)ms$	$2e + (\|\Delta\| + 2)E_2 \approx (11.596 + 0.712\|\Delta\|)ms$
Our scheme	$(d + 1)E_1 + 2E_2 \approx (2.118 + 0.694d)ms$	$2e \approx 10.172ms$

2) Conditional Identity Privacy: The sensors are identified by a set of sequences $\{SN_1, \ldots, SN_n\}$ assigned by TTP, while the encryption of message is only associated with a set of attributes $\gamma \in U$. Thus, the applications cannot know which sensor node the ciphertext comes from and other identity information, which guarantees the privacy of the sensor node identity [43].

3) Fine-Grained Access Control: In this proposed scheme, users can customize a series of access rules to limit the access of the application to system resources. TTP makes the decryption key available only to the applications that meets the access structure ($AS = 1$) by generating the relevant access tree *AS*, thus achieving the personalized fine-grained access control.

4) Collusion Attack: In LSSS, collusion is necessary but not avoided. However, the proposed scheme can resist the collusion attack between unauthorized applications. Suppose there are two malicious applications *A* and *B* that meet only partial access conditions, each of them is unable to obtain the key, and thus attempt to collude. But the proposed scheme can resist this collusion attack. In our KP-ABE scheme, the key is only generated by TTP through the access tree and distributed to the application that complies with all the requirements. So our scheme does not allow unauthorized applications to collude to bypass the access policy to obtain the key and illegally access-related resources.

VII. ANALYSIS OF COMPUTATION AND COMMUNICATION COSTS

Herein, we analyze the cost of the proposed scheme; and also show the results of comparison with other schemes in Tables II and III.

Notations of related operations and some letters are defined as follows.

1) *Q:* Nodes that satisfy the access structure.
2) *E_1:* Exponentiation operation in group G_1. The cost (time) of this operation is about 0.694 ms.
3) *E_2:* Exponentiation or multiplication in group G_2. The cost (time) of this operation is about 0.712 ms.
4) *e:* Bilinear pairing operation. The cost (time) of this operation is about 5.086 ms.
5) *H(∗):* Hash function operation. The cost (time) of this operation is about 0.001 ms.
6) *L∗:* The bit length of element in *.

A. Communication Cost Analysis

We analyze the communication cost of our ABE in each phase and contrast with other schemes, then list the results in Table II.

Specifically, the public parameter *PK* in the proposed ABE scheme is $O(1)$ group elements, then we can compute the communication cost of *PK*, *MK* as $3L_{E_1} + L_{E_2} + 2L_H$ and L_{Z_p}, so the proposed scheme can be extended to large universe. When generating a key, TTP needs to distribute the key to an application that meets the access structure, the overhead of this phase is $2L_{E_1}$. To access system resources, it needs to get the key first, then to decrypt the corresponding data. So the cost of sending a ciphertext is $(d + 1)L_{E_1} + L_{E_2}$.

Goyal *et al.* [22] adopted the Lagrange polynomial interpolation method in decryption, and the size of public parameters *PK* is linear with the size of attributes universe. The communication cost of *PK*, *MK* is $nL_{E_1} + L_{E_2}$ and $(n + 1)L_{E_1} + L_{Z_p}$. Bethencourt *et al.* [23] proposed a CP-ABE scheme. In their scheme, the size of ciphertext includes two group elements for each leaf node of the access tree and the decryption key consists of two group elements for every attribute given to the application. So the size of the key is big, we can represent the cost of sending a key as $(2n + 1)L_{E_1}$. Construction in [24] can handle any access formula involving AND, OR, NOT, and threshold operations based on nonmonotonic access structures. It also used the Lagrange polynomial interpolation method, so the cost of communication is big. In [27], the private key added the "helper values" whose size is increased by an element of $|\Gamma|$, where Γ is the distinct attribute set used in the step of generating the private key.

Above all, as can be seen from the comparison results in Table II, our scheme has a lower communication overhead on the basis of achieving the same security, and is more suitable for the internal environment of unmanned vehicles.

B. Computation Cost

The proposed scheme is established on LSSS. Here, we only introduce our scheme in detail, and other related solutions can

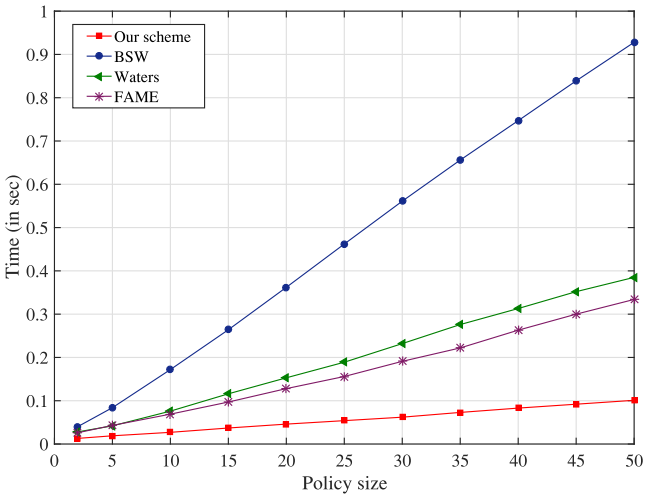


Fig. 6. Encryption time increases proportionally with the policy size.

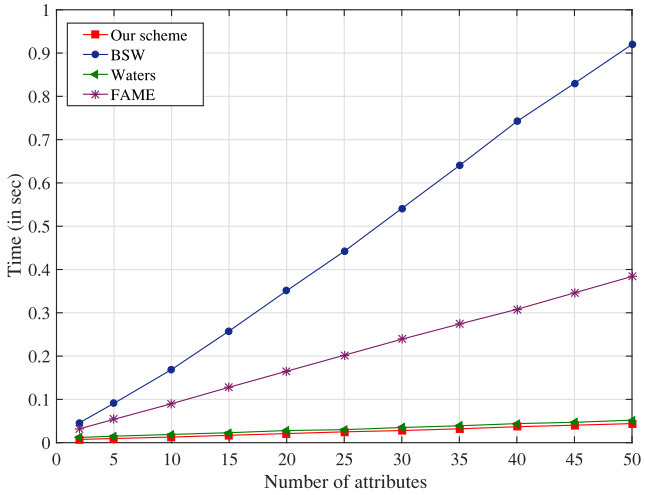


Fig. 7. Key generation time increases proportionally with the number of attributes.

be analyzed in the same way. The comparison of computation cost is shown in Table III.

The computation cost of encryption and decryption algorithms are both simple. In the encryption phase, our algorithm will require one exponentiation for each attribute associated with the ciphertext and one exponentiation, one multiplication in group G_2, hence the cost is $(d+1)E_1 + 2E_2 \approx (2.118 + 0.694d)$ ms. Then, two pairings are required during the decryption phase in the proposed scheme.

VIII. Implementation and Evaluation

We implement a prototype of the proposed scheme in Python 3.7.3 under Ubuntu 19.04 and the performance is measured on a desktop machine running Intel Core i5-7500 3.40-GHz processor with 16-GB RAM. We carry out simulation experiments in two parts. In one part, we implement the proposed ABE scheme and compare its performance with other schemes based on Charm library (version 0.50b) [44], using Type-III curve MNT224 for pairings that takes advantage of asymmetric groups to generate a bilinear map $e : G_1 \times G_2 \rightarrow G_T$. And it can provide 96-b security level [45]. The other part is that we encrypt the online and offline video data with the hybrid encryption method. We first use AES (128 b in the CBC mode) to encrypt the data, then the key of this symmetric encryption $Key_{-\mathrm{AES}}$ is encrypted by the proposed ABE algorithm $Enc(Key_{-\mathrm{AES}}, \gamma)$, lastly only the applications satisfying the policy can decrypt according data. In this way, we can realize the proposed in-vehicle data access control scheme and make a corresponding performance evaluation. For accuracy, we repeat our implementation 20 times on a desktop machine for each scheme, then the average value is taken as the result of each experiment and used to draw the contrast diagram.

A. Performance Analysis of the ABE Scheme

Besides our ABE scheme, we implement BSW's [23] scheme, Waters's [46] scheme, and Agrawal's FAME [47] under the same setting. As we know, KP-ABE and CP-ABE are opposite symmetric solutions, so we transform the proposed scheme into CP-ABE and make a comparative analysis with the above classical schemes. BSW formally defines CP-ABE and it is widely used in later theoretical studies. Waters's [46] scheme is faster with a remarkable performance, also more comparative. FAME is a fully secure ABE scheme based on a standard assumption. We compare the above schemes from three aspects: 1) encryption time; 2) key generation time; and 3) decryption time.

We adopt an access policy of form {Attr_1 and Attr_2 and $\cdots$ and Attr_n}, the size of the policy is n. In our implementation, we set the size of the policies from 0 to 50 to test all the schemes, then we can get 50 different policies. These policies are converted to the access tree to generate decryption keys by TTP. The number of attributes is also set from 0 to 50.

As described above, we develop 50 access policies and analyze the change of scheme cost as the number of access policies increasing, the experimental results are shown in Fig. 6. We can observe that the cost of our encryption is smaller than others, and it takes almost 0.1 s with policy size 50. Furthermore, we show that the key generation time increases proportionally with the number of attributes in Fig. 7. What is more, Waters's and the proposed scheme are superior to other schemes. Fig. 8 illustrates the decryption time cost of all schemes. It is obvious that our scheme and FAME outperform the others, which only take about dozens of milliseconds to decrypt. BSW's and Waters's decryption time both increase in direct proportion to the number of attributes. Therefore, these schemes are not suitable for the situation where there is a large universe of attributes.

According to the experiment, we assume that the size of the attribute sets and the policy are both 50. Then, the complete implementation of the proposed ABE scheme, including set-up, encryption, key generation, and decryption algorithm, takes about 0.54 s. It is worth noting that when applying the ABE algorithm to our access control scheme, we only need to update the key periodically or when an application is deleted or installed. Thus, the actual overhead due to adding attribute encryption is smaller. In the next part, we will introduce the actual cost of the simulation experiment.

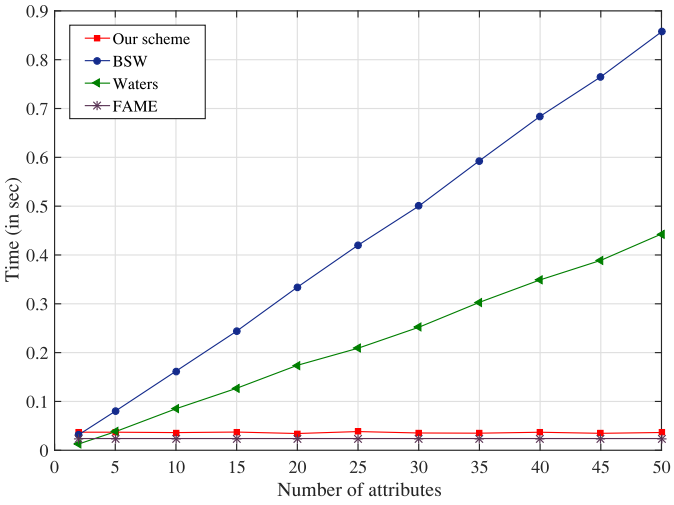


Fig. 8. Change of decryption time cost as the number of attributes increased.

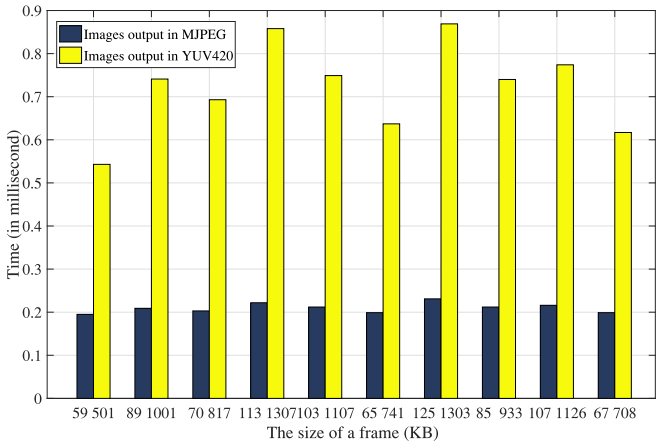


Fig. 9. Cost of frame encryption.

B. Evaluation of Simulation Experiment

In this section, we show the performance tradeoff to simulate the proposed access control scheme in CAVs. The proposed scheme leverages AES and ABE to encrypt both online and offline types of video data. Herein, we encrypt video on the desktop to simulate the encryption of perceived video information when the access control system is applied on CAVs. We consider online video as the realtime data perceived by the camera and the offline video as the important historical data stored in the vehicle.

When simulating the encryption of real-time video in vehicle, our experimental scheme is tantamount to capture video into video frames. Each frame is equivalent to the real-time raw data in the CAV, then we use the AES algorithm to encrypt every video frame. In the implementation, we use two encoding methods MJPEG and YUV420 to process video frames, and we set FPS as 30. First, the above video is intercepted into frames encoded as MJPEG, and the AES algorithm is carried out frame by frame. Then, we repeat the process, except that we change the frame encoding format to YUV420. Finally, we cut off ten consecutive frames and the comparison results of the two encryption experiments are shown in Fig. 9. We can point out that the size of the frame for the same video frame

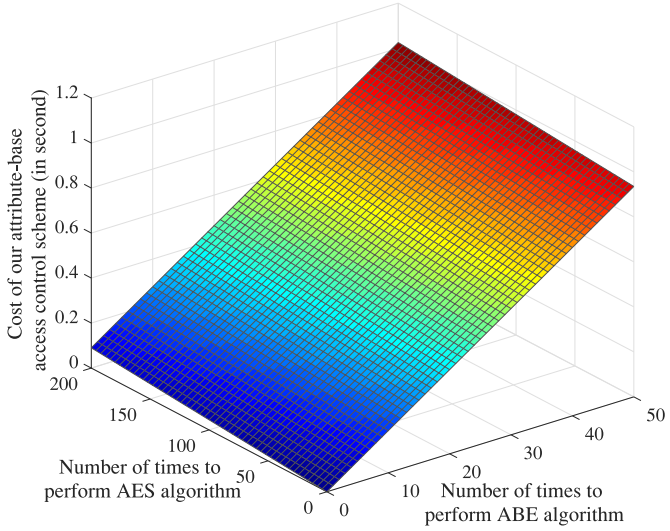


Fig. 10. Cost of the whole access control scheme.

YUV420 was about ten times larger than that for the MJPEG frame. Even so, the overhead of using AES for encryption is very small, at the microsecond level.

When encrypting offline video, we consider a video in format MP4 as a file, and find that the cost of encrypting this type of file is also at the millisecond level.

Remarkably, when implementing the proposed access control scheme, the ABE's set-up algorithm needs to be executed only once during initialization, and the key distributed by TTP can be reused by the application during the period without a key update. Therefore, the cost of the whole scheme mainly consists of the time of implementing the encryption, decryption of ABE and taking AES algorithm as shown in Fig. 10. (In the worst case, every time the AES algorithm is executed, and the ABE algorithm is executed.) In fact, there is no need to execute the ABE algorithm all the time, because the key of AES can be reused by applications to decrypt the data. So the overhead of the proposed scheme only includes the cost of AES, the delay is at the level of microseconds, and it is suitable for an in-vehicle environment.

In a word, the experiment proves that the proposed attribute-based access control scheme improves the security and privacy of CAVs to a large extent, and the whole scheme cost is acceptable.

IX. Conclusion and Future Work

In this study, we proposed an attributed-based access control scheme to fine-grain manage upper applications' access to in-vehicle data. This scheme is built on the distributed structure of CAVs and adopts a fast ABE that allows only the applications that meet the user-defined access policies to obtain corresponding information. Detailed security proof and analysis reveal that the proposed scheme can achieve its multiple security goals. We also proved that the proposed ABE scheme has a better performance than other security schemes. Finally, its implementation shows that the cost of the proposed scheme is relatively small, as its functions perform on the microsecond level.

In future work, we will continue to study the security and privacy of CAVs, aiming to use more lightweight algorithms with attribute revocation and policy hiding to meet more complex security design requirements.

REFERENCES

[1] S. Liu, J. Tang, Z. Zhang, and J.-L. Gaudiot, "Computer architectures for autonomous driving," *Computer*, vol. 50, no. 8, pp. 18–25, 2017.

[2] Q. Zhang *et al.*, "OpenVDAP: An open vehicular data analytics platform for CAVs," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2018, pp. 1310–1320.

[3] A. M. Wyglinski, X. Huang, T. Padir, L. Lai, T. R. Eisenbarth, and K. Venkatasubramanian, "Security of autonomous systems employing embedded computing and sensors," *IEEE Micro*, vol. 33, no. 1, pp. 80–86, Jan./Feb. 2013.

[4] M. Quigley *et al.*, "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw.*, vol. 3. Kobe, Japan, 2009, p. 5.

[5] W. Zeng, M. A. S. Khalid, and S. Chowdhury, "In-vehicle networks outlook: Achievements and challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1552–1571, 3rd Quart., 2016.

[6] J. Liu, S. Zhang, W. Sun, and Y. Shi, "In-vehicle network attacks and countermeasures: Challenges and future directions," *IEEE Netw.*, vol. 31, no. 5, pp. 50–58, Sep. 2017.

[7] Z. Lu, Q. Wang, X. Chen, G. Qu, Y. Lyu, and Z. Liu, "LEAP: A lightweight encryption and authentication protocol for in-vehicle communications," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, 2019, pp. 1158–1164.

[8] F. Martín, E. Soriano, and J. M. Cañas, "Quantitative analysis of security in distributed robotic frameworks," *Robot. Auton. Syst.*, vol. 100, pp. 95–107, Feb. 2018.

[9] X. Xu and C.-K. Fan, "Autonomous vehicles, risk perceptions and insurance demand: An individual survey in China," *Transp. Res. A Policy Pract.*, vol. 124, pp. 549–556, Jun. 2019.

[10] M. Maurer, J. C. Gerdes, B. Lenz, and H. Winner, *Autonomous Driving: Technical, Legal and Social Aspects*. Berlin, Germany: Springer Nat., 2016, pp. 1–7.

[11] W. Enck *et al.*, "TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones," *ACM Trans. Comput. Syst.*, vol. 32, no. 2, p. 5, 2014.

[12] B. Shebaro, O. Oluwatimi, and E. Bertino, "Context-based access control systems for mobile devices," *IEEE Trans. Depend. Secure Comput.*, vol. 12, no. 2, pp. 150–163, Mar./Apr. 2015.

[13] P. Koopman and M. Wagner, "Autonomous vehicle safety: An interdisciplinary challenge," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 1, pp. 90–96, Jan. 2017.

[14] K. Jo, J. Kim, D. Kim, C. Jang, and M. Sunwoo, "Development of autonomous car—Part I: Distributed system architecture and development process," *IEEE Trans. Ind. Electron.*, vol. 61, no. 12, pp. 7131–7140, Dec. 2014.

[15] K. Jo, J. Kim, D. Kim, C. Jang, and M. Sunwoo, "Development of autonomous car—Part II: A case study on the implementation of an autonomous driving system based on distributed architecture," *IEEE Trans. Ind. Electron.*, vol. 62, no. 8, pp. 5119–5132, Aug. 2015.

[16] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," *J. Guid. Control Dyn.*, vol. 25, no. 1, pp. 116–129, 2002.

[17] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 1, pp. 187–210, Jan. 2018.

[18] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2015.

[19] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3061–3070.

[20] K. J. Kim, P. Sundaram, A. H. Leutheuser, and U. P. Mudalige, "Obstacle avoidance co-pilot for autonomous vehicles," U.S. Patent App. 15 240 108, Feb. 2018.

[21] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2005, pp. 457–473.

[22] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Security*, 2006, pp. 89–98.

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

[24] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proc. 14th ACM Conf. Comput. Commun. Security*, 2007, pp. 195–203.

[25] A. Lewko, A. Sahai, and B. Waters, "Revocation systems with very small private keys," in *Proc. IEEE Symp. Security Privacy*, 2010, pp. 273–285.

[26] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Proc. Annu. Int. Cryptol. Conf.*, 2005, pp. 258–275.

[27] S. Hohenberger and B. Waters, "Attribute-based encryption with fast decryption," in *Proc. Int. Workshop Public Key Cryptography*, 2013, pp. 162–179.

[28] X. Mao, J. Lai, Q. Mei, K. Chen, and J. Weng, "Generic and efficient constructions of attribute-based encryption with verifiable outsourced decryption," *IEEE Trans. Depend. Secure Comput.*, vol. 13, no. 5, pp. 533–546, Sep./Oct. 2016.

[29] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM Comput. Surveys*, vol. 35, no. 2, pp. 114–131, 2003.

[30] P. Pal, G. Lauer, J. Khoury, N. Hoff, and J. Loyall, "P3S: A privacy preserving publish-subscribe middleware," in *Proc. 13th Int. Middleware Conf.*, 2012, pp. 476–495.

[31] D. Thatmann, S. Zickau, A. Förster, and A. Küpper, "Applying attribute-based encryption on publish subscribe messaging patterns for the Internet of Things," in *Proc. IEEE Int. Conf. Data Sci. Data Intensive Syst.*, 2015, pp. 556–563.

[32] K. Yang, K. Zhang, X. Jia, M. A. Hasan, and X. S. Shen, "Privacy-preserving attribute-keyword based data publish-subscribe service on cloud platforms," *Inf. Sci.*, vol. 387, pp. 116–131, May 2017.

[33] S. Cui, S. Belguith, P. De Alwis, M. R. Asghar, and G. Russello, "Collusion defender: Preserving subscribers' privacy in publish and subscribe systems," *IEEE Trans. Depend. Secure Comput.*, early access, Feb. 13, 2019, doi: 10.1109/TDSC.2019.2898827.

[34] A. Beimel, *Secure Schemes for Secret Sharing and Key Distribution*. Faculty Comput. Sci., Technion–Israel Inst. Technol., Haifa, Israel, 1996.

[35] D. Boneh and X. Boyen, "Efficient selective-ID secure identity-based encryption without random oracles," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2004, pp. 223–238.

[36] L.-Y. Yeh and J.-L. Huang, "PBS: A portable billing scheme with fine-grained access control for service-oriented vehicular networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 11, pp. 2606–2619, Nov. 2014.

[37] L.-Y. Yeh, Y.-C. Chen, and J.-L. Huang, "ABACS: An attribute-based access control system for emergency services over vehicular ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 3, pp. 630–643, Mar. 2011.

[38] H. Zhang, J. Wang, and J. Chang, "An access control model for multi-level security in multi-domain networking environments," in *Proc. IEEE 9th Int. Conf. Model. Identification Control (ICMIC)*, 2017, pp. 809–814.

[39] X. Zheng, C.-T. Huang, and M. Matthews, "Chinese remainder theorem based group key management," in *Proc. ACM 45th Annu. Southeast Reg. Conf.*, 2007, pp. 266–271.

[40] B. Sun, Q. Li, and B. Tian, "Local dynamic key management scheme based on layer-cluster topology in WSN," *Wireless Pers. Commun.*, vol. 103, no. 1, pp. 699–714, 2018.

[41] J. Zhang, J. Cui, H. Zhong, Z. Chen, and L. Liu, "PA-CRT: Chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular ad-hoc networks," *IEEE Trans. Depend. Secure Comput.*, early access, Mar. 11, 2019, doi: 10.1109/TDSC.2019.2904274.

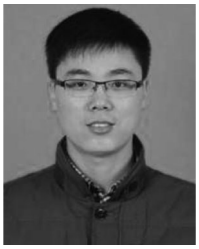
[42] M. A. Alazzawi, K. Chen, A. A. Yassin, H. Lu, and F. Abedi, "Authentication and revocation scheme for VANETs based on Chinese remainder theorem," in *Proc. HPCC/SmartCity/DSS*, 2019, pp. 1541–1547.

[43] J. Cui, X. Zhang, H. Zhong, J. Zhang, and L. Liu, "Extensible conditional privacy protection authentication scheme for secure vehicular networks in a multi-cloud environment," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1654–1667, 2020, doi: 10.1109/TIFS.2019.2946933.

[44] J. A. Akinyele *et al.*, "CHARM: A framework for rapidly prototyping cryptosystems," *J. Cryptograph. Eng.*, vol. 3, no. 2, pp. 111–128, 2013.

[45] B. Yang, K. Yang, Y. Qin, Z. Zhang, and D. Feng, "DAA-TZ: An efficient DAA scheme for mobile devices using arm trustzone," in *Proc. Int. Conf. Trust Trustworthy Comput.*, 2015, pp. 209–227.

[46] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proc. Int. Workshop Public Key Cryptography*, 2011, pp. 53–70.
[47] S. Agrawal and M. Chase, "FAME: Fast attribute-based message encryption," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2017, pp. 665–682.



Jie Cui (Member, IEEE) was born in Henan, China, in 1980. He received the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2012.

He is currently a Professor and a Ph.D. Supervisor with the School of Computer Science and Technology, Anhui University, Hefei. He has over 100 scientific publications in reputable journals, such as IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, academic books, and international conferences. His current research interests include applied cryptography, IoT security, vehicular *ad hoc* network, cloud computing security, and software-defined networking.

Xuelian Chen is currently pursuing the master's degree with the School of Computer Science and Technology, Anhui University, Hefei, China.

Her research focuses on security in connected and autonomous vehicles.

Jing Zhang (Graduate Student Member, IEEE) is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Anhui University, Hefei, China.

She has nearly 20 scientific publications in reputable journals, such as IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, *Information Sciences*, *Science China Information Sciences*, and *Vehicular Communications* and international conferences. Her research interests include vehicular *ad hoc* network, IoT security, and applied cryptography.

Qingyang Zhang (Member, IEEE) received the B.Eng. degree in computer science and technology from Anhui University, Hefei, China, in 2014, where he is currently pursuing the Ph.D. degree.

His research interests include edge computing, computer systems, and security.

Hong Zhong (Member, IEEE) was born in Anhui, China, in 1965. She received the Ph.D. degree in computer science from the University of Science and Technology of China, Hefei, China, in 2005.

She is currently a Professor and a Ph.D. Supervisor with the School of Computer Science and Technology, Anhui University, Hefei. She has over 120 scientific publications in reputable journals, such as IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, and IEEE TRANSACTIONS ON BIG DATA, academic books, and international conferences. Her research interests include applied cryptography, IoT security, vehicular *ad hoc* network, cloud computing security, and software-defined networking.