Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 20(12): 20809–20827.
DOI: 10.3934/mbe.2023921
Received: 12 September 2023
Revised: 06 November 2023
Accepted: 13 November 2023
Published: 20 November 2023

Research article

Message sharing scheme based on edge computing in IoV

Shufen Niu, Wei Liu*, Sen Yan and Qi Liu

College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China

* **Correspondence:** Email: 836602482@qq.com.

Abstract: With the rapid development of 5G wireless communication and sensing technology, the Internet of Vehicles (IoV) will establish a widespread network between vehicles and roadside infrastructure. The collected road information is transferred to the cloud server with the assistance of roadside infrastructure, where it is stored and made available to other vehicles as a resource. However, in an open cloud environment, message confidentiality and vehicle identity privacy are severely compromised, and current attribute-based encryption algorithms still burden vehicles with large computational costs. In order to resolve these issues, we propose a message-sharing scheme in IoV based on edge computing. To start, we utilize attribute-based encryption techniques to protect the communications being delivered. We introduce edge computing, in which the vehicle outsources some operations in encryption and decryption to roadside units to reduce the vehicle's computational load. Second, to guarantee the integrity of the message and the security of the vehicle identity, we utilize anonymous identity-based signature technology. At the same time, we can batch verify the message, which further reduces the time and transmission of verifying a large number of message signatures. Based on the computational Diffie-Hellman problem, it is demonstrated that the proposed scheme is secure under the random oracle model. Finally, the performance analysis results show that our work is more computationally efficient compared to existing schemes and is more suitable for actual vehicle networking.

Keywords: Internet of Vehicles (IoV); attribute-based encryption; authentication; batch verification; edge computing

1. Introduction

With the development of science and technology and the improvement of economic level, people have higher requirements for the convenience and safety of the transportation system. As a kind of mobile ad hoc networks, vehicular ad hoc networks (VANETs) play an important role in traffic management and congestion control. In a VANETs environment, each vehicle acts as a node, allowing vehicle-to-vehicle communication. For example, on a road where an accident occurs, vehicles can alert each other of alternative routes to avoid traffic congestion caused by the accident.

The Internet of Things [1] has accelerated the transition from traditional VANETs to the IoV, thanks to quickly developing 5G and wireless network communication technologies. Through equipment like vehicle units, roadside acquisition modules, and vehicle-road communication units, the IoV makes it possible to collect vehicle operation data in real-time. It creates a data platform for continuously monitoring the real-time operation data of large-scale vehicles and offers a variety of data services. On-Board Units and Roadside Units make up the majority of the Internet of Vehicles [2, 3]. Each vehicle can communicate wirelessly with the infrastructure using the OBU and DRSC protocols [4–7] and broadcast traffic data to other connected vehicles via the RSU. Depending on the message it receives, the vehicle will select the best route [8]. The Internet of Vehicles can take advantage of cloud computing to improve the IoV [9], which enables quick communication between vehicles by storing and processing collected messages, but this consumes a lot of computer bandwidth and network resources. The OBU in the vehicle is a low-powered IoT device which is lightweight. Therefore, the development of edge computing has successfully addressed this issue. When the Internet of Vehicles and edge computing are combined, a portion of the vehicle's computing tasks are delegated to the edge service nodes [10], which reduces the user's computing burden, reduces the time it takes for data to be transmitted over networks, and significantly increases the efficiency of data processing [11–13]. Additionally, this complies with the Internet of Vehicles' real-time processing standards. Roadside devices serve as edge nodes in the Internet of Vehicles, assisting vehicles with calculations and accelerating vehicle-to-vehicle communication.

1.1. Motivations and contributions

The Internet of Vehicles plays an important role in ensuring road traffic safety, but its open communication environment also makes the message transmitted in the network have the risk of being eavesdropped and tampered with. Due to the characteristics of the Internet of Vehicles, the receiver of the message has variability, so the ordinary public key encryption scheme cannot meet the security requirements of the Internet of Vehicles. At the same time, in order to protect the privacy of vehicles, the dissemination of information usually needs to be carried out anonymously. However, the existing cloud-based message sharing scheme will lead to higher delay and service corresponding time in the Internet of Vehicles environment, and thus it is not suitable for an environment with limited vehicle computing power.

Thus, we propose a message-sharing scheme based on edge computing in the IoV. The main contributions of the proposed scheme are as follows.

- The attribute-based encryption algorithm is used to encrypt the message, which ensures the location of the information transmission target and the specific type of the vehicle, and realizes the confidentiality of the message and fine-grained access control. Edge computing is introduced and a "cloud-edge-device" three-layer architecture is constructed. Some of the encrypted and decrypted calculations are outsourced to the RSU, thereby reducing the computational cost of the vehicle and realizing the application of the scheme in a resource-constrained device environment.
- This scheme adopts anonymous identity-based message verification and tracking technology, which can verify the legitimacy of a message signature while protecting vehicle identity information. When a traffic accident occurs, a trusted third party can determine the true identity of the

accident vehicle. In addition, we also use batch verification to reduce the total verification time on the RSU as an edge node and improve the efficiency of message authentication.

- Theoretical analysis and numerical results show that the proposed scheme has greater advantages in computational efficiency and communication overhead compared with other current works. Therefore, our scheme is effective and more suitable for an IoV environment with more resource-constrained devices.

1.2. Organization

The organization of our paper is shown below. Section 2 introduces existing related works and the novelty of this scheme. Some preliminaries are provided in Section 3. Section 4 introduces the system model, and gives a formal definition. We discuss the scheme in Section 5. We prove the security of the proposed schemes in Section 6. The performance analysis of the scheme is demonstrated comprehensively in Section 7. Finally, we conclude the paper and discuss future research in Section 8.

2. Related works

Security requirements such as message confidentiality, integrity, and privacy protection [14, 15] are essential in IoV communications. Since cloud servers are semi-trusted, attackers may obtain information such as traffic conditions from disseminated messages. The connected vehicles are then broadcast an erroneous message, which may cause confusion or unexpected situations. On the other hand, in some situations, only the intended receiver should have access to the information that has been transmitted. A traditional access control scheme that enables fine-grained access control to assure message delivery to particular areas or specified types of vehicles was proposed by Bethencourt et al. [16]. As the IoV capacity increases, so does the type of vehicle [17]. Road messages are utilised as attributes in classic access control-based VANETs [18] since vehicles must develop various access control policies.

Cui et al. [19] also released a scheme for attribute-based encryption that permits fine-grained access control for vehicle messages. A trusted authority (TA) distributes the vehicle's attribute keys. A ciphertext policy attribute-based encryption (CP-ABE) scheme was proposed forth by Kang et al. [20] to create a flexible access structure for designated vehicles in IoV communication. However, vehicles require much computation and are unsuitable for the current Internet of Vehicles. [21] proposed an ABE model of edge intelligent IoV parallel outsourcing decryption, which only outsourced part of the decryption without reducing the computational burden of the data owner. Based on CP-ABE, Ahmed et al. [22] introduced a new privacy protection collaboration technique, but it needed several exponents and pair operations, which did not satisfy the current vehicle requirements for efficiency.

Anonymity is essential for privacy protection in IoV communication [23]. An anonymous credential scheme was introduced by Chim et al. [24] to protect vehicle privacy and ensure that the car cannot be connected to either side. Pseudonym-based methods can be applied in a range of vehicular communication system settings and are often efficient and straightforward [25]. Vehicle anonymity can be guaranteed by using signature authentication in IoV [26–28]. Previous systems, however, required preloading of public/private key pairs by the vehicles in order to sign or verify messages, which may impose a significant authentication cost on storage, creation, and verification.

We compare our scheme with the related work in Table 1. In an edge-based vehicle framework, the proposed scheme introduces edge computing and uses RSU as an edge node to undertake part of the decryption operation of the vehicle, which reduces the computational burden of the vehicle and is more suitable for the IoV environment where the computational capability of end devices is limited. In terms of identity privacy protection, the proposed scheme supports vehicle anonymization and only a trusted third party can determine the real identity of the vehicle. In terms of message security, we use attribute-based encryption algorithms to protect the confidentiality of messages and support batch verification of the legitimacy of message signatures, which further reduces the time and transmission cost of verifying a large number of message signatures.

Table 1. Functional comparison.

	Our scheme	Kang et al. [20]	Feng et al. [21]	Saidi et al. [22]
Outsourcing encryption	Yes	No	No	Yes
Outsourcing decryption	Yes	No	Yes	Yes
Anonymity	Yes	No	No	No
Verfication	Yes	Yes	No	No
Batch verfication	Yes	No	No	No

3. Preliminaries

3.1. Bilinear pairing

Let G_1 be a cyclic additive group generated by P, whose order is a prime p, and G_2 be a cyclic multiplicative group of the same order p. $e : G_1 \times G_1 \rightarrow G_2$ is referred to as a bilinear map if it meets the attributes as follows [29]:

- Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$ for any $a, b \in Z_p^*$, $P, Q \in G_1$;
- Non-degeneracy: there exist $P, Q \in G_1$, such that $e(P, Q) \neq 1$;
- Computability: there is an algorithm to calculate $e(P, Q)$ for any $P, Q \in G_1$.

3.2. Computational Diffie-Hellman problem (CDHP) [30]

For a tuple $P, xP, yP \in G$, where P is the generator of G having order q and $x, y \in Z_p^*$, it is infeasible to calculate xyP.

3.3. Access structure

We refer to the access structure in this scheme as an access tree that is utilized to conceal the source data's encryption key. As opposed to the non-leaf node x of $\mathcal{T}$, which stands for a threshold (AND gate or OR gate), each leaf node y of $\mathcal{T}$ has an attribute $att(y)$. We suppose that num_x represents the number of children that of node x, k_x represents its threshold, $parent(x)$ represents the parent of node x that is not the root, and $index(x)$ returns the index of node x. If $k_x = num_x$, then x is an AND gate; if $k_x = 1$, then x is an OR gate. A ciphertext CT is ultimately output. It finally outputs a ciphertext CT. R serves as the root node of the access tree $\mathcal{T}$. The expression $\mathcal{T}_x(S) = 1$ can be used to state that an attribute set S satisfies the access tree $\mathcal{T}_x$. It is calculated recursively as follows:

- If x is a non-leaf node, calculate $\mathcal{T}_n(S)$ for all child nodes n of x, and get messages m and $\mathcal{T}_x(S) = 1$ if and only if at least k_x child nodes return.
- If x is a leaf node, then $\mathcal{T}_x(S) = 1$ returns if and only if $att(x) \in S$.

3.4. IoV communication

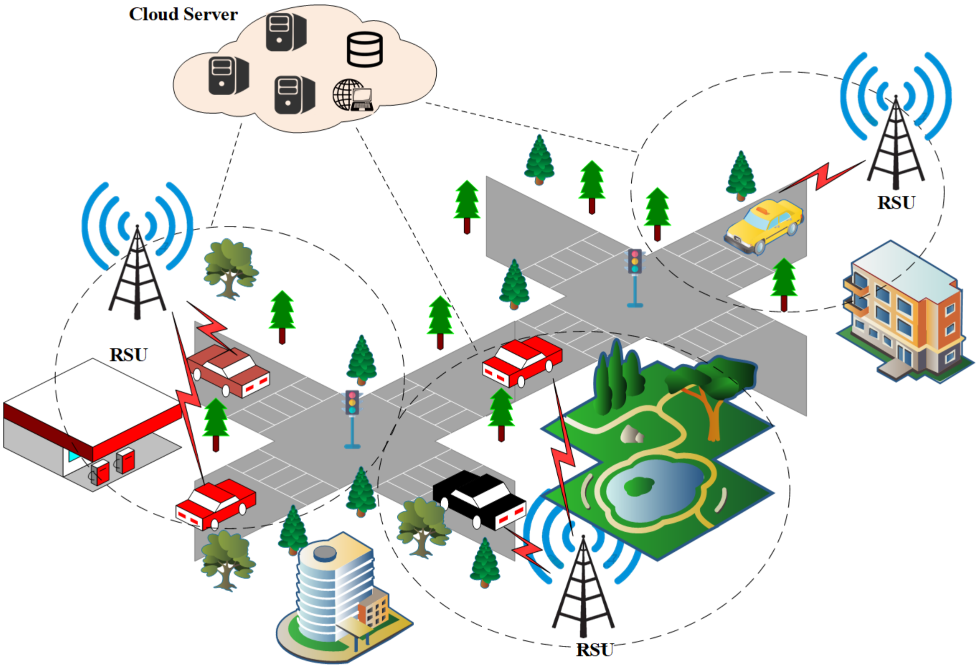


Figure 1. Vehicular network structure.

The Internet of Vehicles allows vehicles to communicate with infrastructure and other vehicles, thereby improving traffic management efficiency and ensuring road safety. The typical scenario of Internet of Vehicles communication is shown in Figure 1, where vehicles on the road share their collected traffic information through RSUs and cloud servers. The communication process is as follows: the sender vehicle completes the message encryption with the assistance of RSU and uploads it to the cloud server. The receiver vehicle initiates a message request to the cloud server. The cloud server sends the message to the RSU nearest the target vehicle for verification and partial decryption. Finally, the partially decrypted ciphertext is given to the receiver vehicle, and the receiver vehicle finally decrypts the message to obtain the message.

4. Scheme overview

4.1. System model

We consider an information-sharing scenario in V2I vehicle networking. As shown in Figure 2, it is composed of five entities. The specific functions of each entity are as follows:

Trusted Authority (TA): The most reputable and authoritative organisation in IoV is TA, the trusted security centre. Public parameters and system master keys must be made available by TA. In

IoV, TA is in charge of not only registering RSUs, but also giving each vehicle a real identity and key.

Cloud Server (CS): Users may get computational and storage services via cloud servers. It receives and stores the ciphertext and signature from the vehicle. After receiving the request sent by vehicle, the CS sends the stored ciphertext and signature to the RSU.

Data Owner (DO): In the IoV, the vehicle that collects and sends road information acts as a DO. It defines the access policy, encrypts the message to generate a complete ciphertext with the help of the RSU, signs the ciphertext with its own anonymous identity, and delivers the ciphertext and signature to the CS.

Data User (DU): In the IoV, the vehicle that wants to obtain road information acts as a DU, makes a request to the cloud server when it satisfies the requirements of the access control policy, and receives data by decrypting part of the ciphertext sent by the RSU.

Roadside Unit (RSU): The RSU is mainly responsible for outsourcing encryption calculation, performing verification calculations on the sent ciphertext, decrypting the ciphertext for users whose attributes satisfy the access policy, and then transmitting partially decrypted data to the corresponding users.

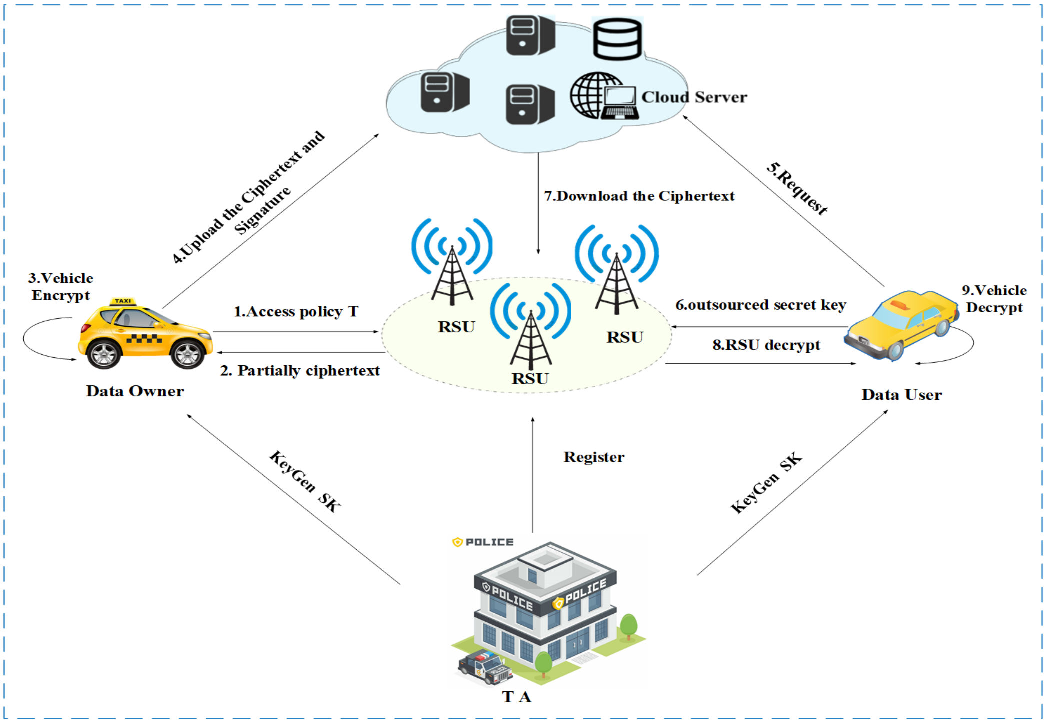


Figure 2. System model.

4.2. *Formal definition*

The scheme consists of the following nine algorithms: Setup, KeyGen, AnonGen, RSU Encrypt, DO Encrypt, Sign, Verify, RSU Decrypt, and DU Decrypt.

Setup: Given security parameters ξ, TA outputs the system public key PK and master key MSK by running this algorithm.

KeyGen: The key generation algorithm inputs attribute set A of the vehicle, public key PK and the master key MSK. The private key SK will be output by the TA.

AnonGen: By entering the real vehicle identity, the TA performs this algorithm to produce the anonymous identity *aid* and anonymous key *ak*.

RSU Encrypt: The RSU runs the encryption algorithm using the access policy $\mathcal{T}$ and the system public key PK, outputs a partially encrypted ciphertext CT' and sends it to the DO.

DO Encrypt: The algorithm is executed by the DO that takes as an input a partially encrypted ciphertext CT', a message m, and the public parameter PK. The DO outputs ciphertext CT to the CS.

Sign: The anonymous identity *aid* and its anonymous key *ak*, a ciphertext CT, a timestamp t, and the public parameter PK are input to the signature algorithm that is run by the DO. It then produces a signature σ.

Verify: An anonymous identity *aid*, a signature σ, a timestamp t, a ciphertext CT and the public parameter PK are all input to the verify algorithm using the RSU. It returns true such that σ is verified.

RSU Decrypt: If the vehicle's characteristics satisfy the policy $\mathcal{T}$, the RSU executes this algorithm, which requires the inputs of a ciphertext CT, an external secret key SK', and PK. It produces a ciphertext CT_0 that is only partially deciphered.

DU Decrypt: The DU performs this decryption algorithm by taking ciphertext CT_0 and SK and outputs a message m.

4.3. Security requirements

Message authentication, identity privacy protection, traceability, unforgeability, confidentiality, and fine-grained access control are some of the security requirements that the scheme should be able to achieve. Details of the above requirements are as follows:

Message authentication: The RSU should be able to confirm whether the message is complete and whether the vehicle transmitting the message is legal once it receives the signature and ciphertext delivered by the CS.

Identity privacy protection: The vehicles remain anonymous since only a trusted third-party TA can determine the real identification of the vehicles.

Traceability: Despite that the vehicle's real identity should be kept a secret from the RSU and other vehicles, the TA can determine the vehicle's real identity from the message it transmitted when malicious vehicles forge messages.

Unforgeability: Malicious vehicles cannot generate fake messages with valid signatures by simulating legitimate vehicles.

Confidentiality: Cloud servers and vehicles' unauthorized access to information cannot obtain the information of vehicles collecting road conditions.

Fine-grained access control: A flexible access policy ought to be employed to ensure the message distribution, and only certain vehicles should receive it.

5. The proposed scheme

The TA first runs the Setup algorithm to generate system parameters, and runs the KeyGen algorithm and the AnonGen algorithm to generate the private key and anonymous identity of the corresponding vehicle. Then, the encryption algorithm (divided into RSU partial outsourcing encryption algorithm and vehicle encryption algorithm) is executed, the signature algorithm is executed and the ciphertext and signature are sent to the cloud server for storage. After the receiver vehicle initiates a message request to the cloud server, the cloud server sends the signature and ciphertext to the RSU nearest the target vehicle. After receiving the signature and ciphertext, the RSU first verifies its legitimacy. Then, the RSU uses the outsourcing key of the receiver's vehicle to perform partial outsourcing decryption and sends the decryption result to the vehicle. Finally, the vehicle is decrypted to obtain M. The specific implementation of the algorithm is shown in Figure 3.

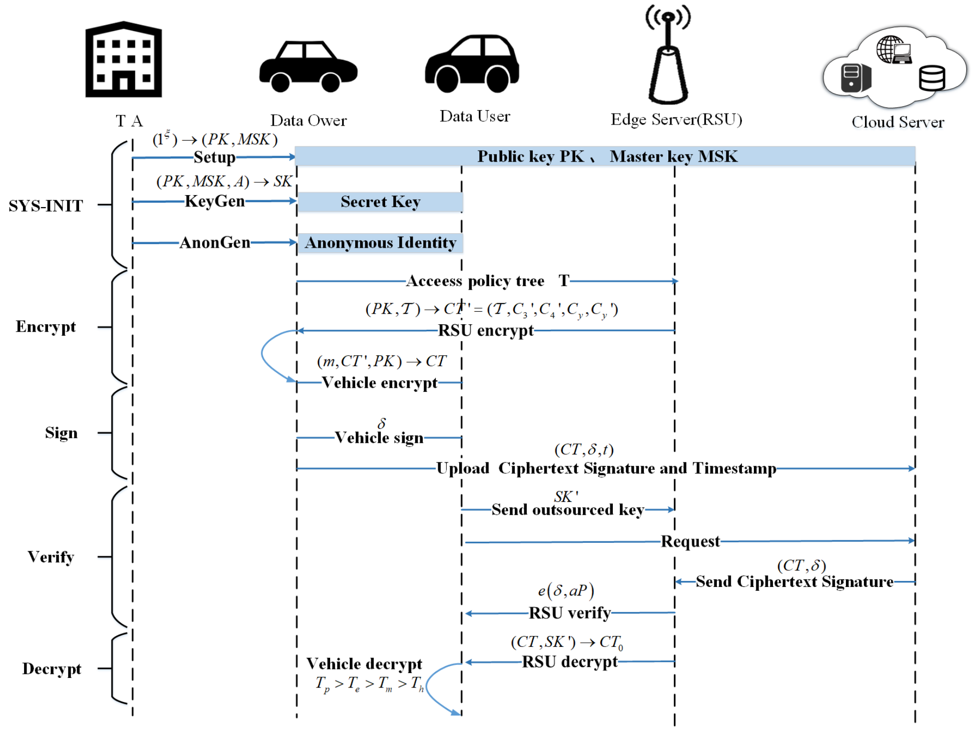


Figure 3. Algorithm model.

- **Setup:** TA runs this algorithm to initialize the system. Define the security parameters 1^k.
    - Provide two cyclic groups, G_1 and G_2, both of prime order p, where G_1 is an additive cyclic group and G_2 is a multiplicative cyclic group. P and Q are randomly chosen by the TA, while $a, b \in Z_p^*$ can be also randomly chosen.
    - Select three hash functions: $H_1 : \{0,1\}^* \rightarrow G_1$, $H_2 : G_1 \rightarrow Z_p^*$, $H_3 : \{0,1\}^* \rightarrow Z_p^*$.
    - Construct a bilinear map $e : G_1 \times G_1 \rightarrow G_2$ and output $e(P,P)^{ab}$. Expose system parameters $PK = (P, Q, aP, bQ, e(P,P)^{ab})$ and retain the system master key $MSK = (a, b)$.

- **KeyGen:** The algorithm is run by the TA. It chooses unique $\alpha, \beta \in Z_p^*$ and selects $r_j \in Z_p^*$ related to each attribute $j \in A$ from the attribute set A randomly. The secret key for the corresponding vehicle is output by the TA.

$$SK = \left\{ \begin{array}{l} D = (b + \beta) P, D_1 = (\beta P + \alpha b Q), D_2 = \alpha a P, \\ D_j = \beta a P + r_j H_1(j), D_j' = r_j P \end{array} \right\}_{j \in A} \tag{5.1}$$

- **AnonGen:** The TA chooses $u_1 \in Z_p^*$ between the RSU and the vehicle and gives the vehicle a true identity rid. The TA generates the vehicle's shared secret u_2 and calculates the verification key as $vk = u_2 \oplus rid$. The verification table will store the (vk, u_1). Then, the vehicle selects $\theta \in Z_p^*$ at random and outputs the anonymous identity $aid = (z_1 = \theta aP, z_2 = vk \oplus H_2(u_1\theta aP))$, where the vehicle maintains the anonymous key $ak = \theta$.

- **Encrypt:** To begin, the vehicle utilizes a random symmetric key $k \in G_2$, to encrypt the message m. The ciphertext can be represented by the notation $C = SE_k(m)$ if the symmetric encryption algorithm is SE. The access policy $\mathcal{T}$ is then uploaded to the RSU by the vehicle. The following is a detailed description of the RSU Encrypt and Vehicle Encrypt processes:

  - **RSU Encrypt:** The outsourced encryption is carried out by the RSU utilizing encryption algorithm. The polynomials are picked in a top-down manner, beginning with the root node R, in accordance with the provided access policy $\mathcal{T}$. The threshold is $k_x = d_x + 1, \forall x \in \mathcal{T}$, if d_x is the degree of q_x. If x is the root node R, the algorithm selects $s \in Z_p^*$ and sets $q_R(0) = s$. When y is the child node of x, it sets $q_y(0) = q_x(index(y))$. We assume Y to be the set of leaf nodes in $\mathcal{T}$. Then, the RSU returns a result CT' that is only partially encrypted as

$$CT' = \left\{ \begin{array}{l} \mathcal{T}, C_3{}' = saP, C_4{}' = sbQ, \\ C_y = q_y(0)P, C_y{}' = q_y(0)H_1(att(y)) \end{array} \right\}_{y \in Y} \tag{5.2}$$

  - **Vehicle Encrypt:** First, the vehicle judges whether $e(C_3{}', bQ) = e(C_4{}', aP)$ and $e\left(C_y, H_1\left(attr_y\right)\right) = e\left(C_y{}', P\right)$ are equal after receiving the partial ciphertext CT'. If the two equations are satisfied, the vehicle runs the encryption algorithm and randomly selects $\varphi \in Z_p^*$. Then, calculate:

$$\left\{ \begin{array}{l} C_1 = k \cdot e(P,P)^{ab\varphi} \\ C_2 = \varphi aP \\ C_3 = C_3{}' + \varphi aP = (s+\varphi)aP \\ C_4 = C_4{}' + \varphi bQ = (s+\varphi)bQ \end{array} \right. \tag{5.3}$$

  And the vehicle outputs ciphertext CT,

$$CT = \left(\begin{array}{l} C = SE_k(m), C_1, C_2, C_3, C_4, \\ \left\{C_y = q_y(0)P, C_y{}' = q_y(0)H_1(att(y))\right\}_{y \in T} \end{array} \right) \tag{5.4}$$

- **Sign:** Under the current timestamp t, the vehicle inputs the anonymous identity aid and its anonymous key ak, a ciphertext CT to calculate $c = H_3(CT \parallel z_1 \parallel z_2 \parallel t)$, and the signature $\delta = (\theta + c)bQ$ of the ciphertext. Finally, the vehicle sends the ciphertext CT, signature δ, and timestamp t to the cloud server for storage.

- **Verify:** To make sure that the vehicle is not attempting to distribute misleading information to others, the RSU first checks the signature after getting the signature and ciphertext from CS. A detailed description of the single verification and batch verification processes follows:

  - **Single Verification:** The RSU runs the algorithm that calculates whether the timestamp t is correct to withstand replaying attacks, checks $vk \oplus H_2(u_1 z_1) = z_2$ and calculates $c = H_3\left(CT \parallel z_1 \parallel z_2 \parallel t\right)$ with the final encrypted message and signature. Verify the validity of the signature by checking whether $e\left(\delta, aP\right) = e\left(z_1 + caP, bQ\right)$ holds.

  - **Batch Verification:** The batch verification algorithm can be used to simultaneously check the final message, when the RSU receives different ciphertexts CT_1, CT_2, $\ldots$, CT_n, and signatures $\delta_1, \delta_2, \ldots, \delta_n$. Allow b to have a low value. First, a random vector $v = \{v_1, v_2 \cdots v_n\}$ is created with the small exponent test by the RSU, which is between 1 and 2^b, to ensure the unforgeability of the signature. The RSU then calculates $c_i = H_3\left(CT_i \parallel z_{i,1} \parallel z_{i,2}\right)$ and $e\left(\sum_{i=1}^{n} v_i \delta_i, aP\right) = e\left(\sum_{i=1}^{n} v_i z_{i,1} + (\sum_{i=1}^{n} v_i c_i) aP, bQ\right)$, if the received message is legal.

- **RSU Decrypt:** The vehicle user requests to access the IoV data stored in the cloud server. $\lambda \in Z_p^*$ is selected , the outsourced key SK' is calculated, and it is sent to the RSU by the vehicle. When the vehicle attributes satisfy the access policy $\mathcal{T}$, the CS sends the ciphertext to the RSU, which can help the vehicle decrypt the message, and then the final ciphertext is sent to the vehicle so that the vehicle can decrypt it by itself.

$$SK' = \begin{pmatrix} D' = \lambda D, D_1{}' = \lambda D_1, D_2{}' = \lambda D_2, \\ \left\{E_j = \lambda D_j, E_j{}' = \lambda D_j{}'\right\}_{j \in A} \end{pmatrix} \tag{5.5}$$

The RSU utilizes SK' to execute the decryption algorithm after receiving CT. If z is all the children nodes of x, $DecryptNode\left(CT, SK', z\right)$ is run by the RSU, and the output is stored as F_Z. x is a input for $DecryptNode\left(CT, SK', x\right)$. If x is a leaf node, $i = attr_x$. If $i \in A$, we have

$$DecryptNode\left(CT, SK', x\right) = e(P, P)^{\lambda \beta a q_x(0)} = F_x \tag{5.6}$$

If $i \notin A$, $DecryptNode\left(CT, SK', x\right) = \perp$ is output. S_x is referred to as a child node of order x when $F_z \neq \perp$. Let $S_x{}' = index\left(z : z \in S_x\right)$ and $j = index\left(z\right)$. In the end, it calculates

$$\begin{aligned} F_x &= \prod_{z \in_x} F_z{}^{\Delta_{j,S_x{}'}(0)} \\ &= \prod_{z \in_x} (e(P, P)^{\lambda \beta a q_x(j) \Delta_{j,S_x{}'}(0)} \\ &= e(P, P)^{\lambda \beta a q_x(0)} \end{aligned} \tag{5.7}$$

When every single attribute $\mathcal{T}$ on the attribute tree is fully satisfied, then

$$\begin{aligned} F_R &= DecryptNode\left(CT, SK', R\right) \\ &= e(P, P)^{\lambda \beta a q_R(0)} \\ &= e(P, P)^{\lambda \beta a s} \end{aligned} \tag{5.8}$$

After that, the RSU calculates the following:

$$\begin{aligned} B &= \frac{e\left(D_1{}', C_3\right)}{e\left(D_2{}', C_4\right)} = \frac{e\left(\lambda\left(\beta P + \alpha bQ\right), (s+\varphi) aP\right)}{e\left(\lambda\alpha aP, (s+\varphi) bQ\right)} \\ &= \frac{e\left(\lambda\beta P, (s+\varphi) aP\right) \cdot e\left(\lambda\alpha bQ, (s+\varphi) aP\right)}{e\left(\lambda\alpha aP, (s+\varphi) bQ\right)} \\ &= (P, P)^{\lambda\beta a(s+\varphi)} \end{aligned} \tag{5.9}$$

and

$$\begin{aligned} M &= \tfrac{B}{F_R \cdot e(C_2, D')} \\ &= \frac{e(P,P)^{\lambda\beta a(s+\varphi)}}{e(P,P)^{\lambda\beta as} \cdot e\left(\varphi aP, \lambda\left(b+\beta\right) P\right)} \\ &= \frac{1}{e(P,P)^{\lambda a\varphi b}} \end{aligned} \tag{5.10}$$

As a result, the RSU sends the partially decrypted ciphertext $CT_0 = (C, C_1, M)$ to the appropriate vehicle if it satisfies all of the attribute requirements.

- **Vehicle Decrypt:** After running the decryption algorithm to decrypt the symmetric key k using a random number λ when it receives CT_0, the vehicle then uses the secret key SK to return the decrypted symmetric key k.

$$\begin{aligned} k &= C_1 \cdot (M)^{\frac{1}{\lambda}} \\ &= k \cdot e(P,P)^{ab\varphi} \cdot \left(\tfrac{1}{e(P,P)^{\lambda ab\varphi}}\right)^{\frac{1}{\lambda}} \\ &= k \end{aligned} \tag{5.11}$$

The vehicle then uses the encryption algorithm $C = SE_k(m)$ to decrypt the plaintext corresponding to the ciphertext after decrypting k and obtaining it.

6. Security analysis

In this section, we conduct a detailed security analysis of our scheme concerning the security requirements.

6.1. Unforgeability

Theorem 1. Under the random oracle model, we prove that the scheme is unforgeable against adaptive chosen message attacks, assuming that the CDHP problem is difficult.

Proof: We will design challenger C to solve the CDHP by using A, who is an adversary. A CDHP challenge (P, xP, yP) for $x, y \in Z_q^*$, and $P \in G$ is taken by C. By carrying out the following steps, we demonstrate how C uses A to calculate the presented CDHP example xyP.

h-Oracle: Assume that A is unable to determine how to use the the hash function $h(\cdot)$. C keeps an initially empty list h^{list} to respond to h queries. When A queries C with a message (aid_i, m_i, t_i), C replies as follows: If the query (aid_i, t_i) already exists in a tuple (aid_i, m_i, t_i, h_i) on the h^{list} , C returns $h_i = h\left(m_i \parallel t_i \parallel z_{i,1} \parallel z_{i,2}\right)$. In the event that this does not occur, C selects a random number $h_i \in Z_q^*$ to set $h_i = h\left(m_i \parallel t_i \parallel z_{i,1} \parallel z_{i,2}\right)$ and returns h_i to A. The (aid_i, m_i, t_i, h_i) is added to the h^{list}.

Sign Oracle: Even though C does not know the private key, it can nonetheless produce the signature after receiving a signing query for the message m_i. At random, it chooses $r_i, h_i \in Z_q^*$ and $z_{i,2} \in G$. Then, the signature as $S_i \in r_i Q_1$ and $z_{i,1} = r_i P - h_i Q$ is calculated. Checking the following example shows whether $\{aid_i, m_i, S_i, t_i\}$ is a valid signature: $e\left(z_{i,1} + h\left(m_i \parallel t_i \parallel z_{i,1} \parallel z_{i,2}\right) Q, Q_1\right) = e\left(S_i, P\right)$. If the (aid_i, m_i, t_i, h_i) already shows up in h^{list}, C chooses another $r_i, h_i \in Z_q^*$ and $z_{i,2} \in G$, and keeps trying once more. After that, C goes back to A with $\{aid_i, m_i, S_i, t_i\}$, and (aid_i, m_i, t_i, h_i) is stored in the h^{list}. All of C's signatures are identical to those collected by A's legitimate vehicle and are therefore all illegible.

Output: By the Forking Lemma [28], after replaying A with the same random tape, C receives two valid signatures $\left\{aid_i, m_i^*, S_i, t_i\right\}$ and $\left\{aid_i, m_i^*, S_i^*, t_i^*\right\}$ within a polynomial time, where $S_i = (r_i + xh_i) Q_1$, $S_i^* = \left(r_i + xh_i^*\right) Q_1$. Then, C calculates $\left(h_i - h_i^*\right)^{-1} \cdot \left(S_i - S_i^*\right) = xyP$. The final result from C is xyP, which is the CDHP instance's solution.

We demonstrate that C solves the specified instance of the CDHP to complete the proof, although this goes against the notion that the CDHP is difficult. The implication is that an attacker cannot deceive an RSU or a vehicle by forging its signature. It is possible to achieve message authentication, integrity, and non-repudiation in this way.

6.2. *Identity privacy preserving*

We believe it would be difficult for an adversary to discover the true identity of the vehicle. The proposed scheme generates an anonymous identity $aid_i = (z_{i,1}, z_{i,2})$ for the vehicle based on its real identity rid, where $z_{i,1} = r_i P$ and $z_{i,2} = rid \oplus h\left(r_i Q\right)$. An adversary is capable of computing $r_i Q$ or $xz_{i,1}$, if it is aware of the secret r_i from $z_{i,1}$ or the private key x from Q . The adversary can then determine rid by performing the calculations $rid = z_{i,2} \oplus h\left(r_i Q\right)$ or $rid = z_{i,2} \oplus h\left(xz_{i,1}\right)$. We counter that if the CDHP is difficult, the vehicle's anonymous identification cannot expose its true identify. Based on this, our system ensures identity privacy-preserving, meaning that only the TA and the vehicle can determine its true identify.

6.3. *Confidentiality*

The message m is encrypted using a symmetric encryption technique in this scheme, and after that, the confidentiality of the message is equivalent to the confidentiality of a random symmetric key k. From the encryption algorithm, the key for an adversary to break the security of this scheme is to calculate $(P, P)^{ab\varphi}$, which is obviously difficult. Furthermore, even if the outsourced key sent by the vehicle to the RSU is intercepted, it does not reveal any meaningful details about the random number λ used by the vehicle for decryption. In conclusion, neither the cloud server nor an unauthorized vehicle nor a malicious third party can access the information contained in the ciphertext, and our scheme guarantees the confidentiality of the message m.

7. Performance analysis

This section theoretically and numerically analyzes the the advantages and disadvantages of proposed scheme by comparing the schemes of Kang et al. [20], Feng et al. [21] and Saidi et al. [22]. We mainly consider the computational overhead of system initialization, key generation, encryption and decryption. For ease of understanding, Table 2 outlines the symbols used in the theoretical analysis phase. The relationship between the symbol corresponding time overhead is $T_p > T_e > T_m > T_h$.

Table 2. Notations.

Symbols	Description
T_p	Time of pairing operation
T_e	Time of exponential operation
T_m	Time of multiplication operation
T_h	Time of hash operation
n	Number of attributes
$\|G_1\|$	Length of elements in group G_1
$\|G_2\|$	Length of elements in group G_2
$\|Z_r\|$	Length of elements in group Z_p^*

7.1. Theoretical analysis

Table 3. Computation overhead.

	Our scheme	Kang et al. [20]	Feng et al. [21]	Saidi et al. [22]
Setup	$2T_m + T_e + T_p$	$T_m + 2T_e + T_p$	$(2 + n)T_e + T_p$	$3T_e + T_p$
Keygen	$(2n + 8)T_m + nT_h$	$2T_m + 4T_e + T_h$	$(3 + n)T_e$	$2T_m + (4 + 2n)T_e$
EN-enc	$6T_m + nT_h$	——	——	$4T_m + (2 + 7n)T_e + T_p$
DO-enc	$5T_m + T_e + T_p + nT_h$	$2T_m + (4 + n)T_e + T_p$	$(5 + n)T_m + (4 + 3n)T_e + T_p$	$4T_m + (2 + 3n)T_e + T_p$
EN-dec	$(n + 1)T_m + 5T_p$	——	$(1 + n)T_m + 2T_e + T_p$	$(4 + n)T_m + 3nT_e + 5nT_p$
DU-dec	T_m	$(4 + n)T_m + 3nT_e + 5nT_p$	$T_m + T_e$	$T_m + T_e + T_p$

Table 4. Communication overhead.

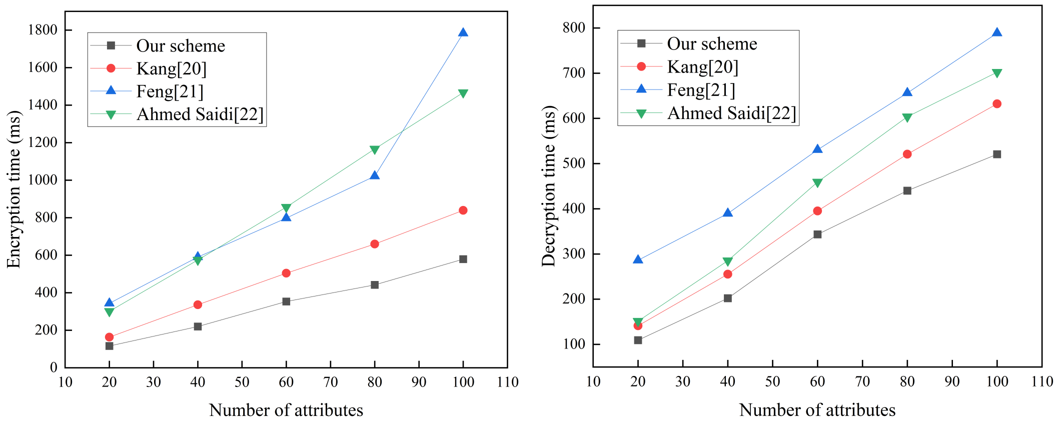
	Our scheme	Kang et al. [20]	Feng et al. [21]	Saidi et al. [22]
Setup	$2\|G_1\| + \|G_2\| + 2\|Z_r\|$	$5\|G_1\| + \|G_2\| + \|Z_r\|$	$(1 + n)\|G_1\| + \|G_2\| + 2\|Z_r\|$	$2\|G_1\| + \|G_2\| + n\|Z_r\|$
Keygen	$(4 + 2n)\|G_1\|$	$(3 + n)\|G_1\|$	$(3 + n)\|G_1\|$	$(2 + 2n)\|G_1\|$
EN-enc	$4\|G_1\|$	——	——	$(3 + 6n)\|G_1\| + \|G_2\|$
DO-enc	$(6 + 2n)\|G_1\| + \|G_2\|$	$(1 + 2n)\|G_1\| + \|G_2\| + (1 + 2n)\|Z_r\|$	$(3 + 3n)\|G_1\| + \|G_2\| + \|Z_r\|$	$(2 + 5n)\|G_1\| + \|G_2\| + \|Z_r\|$
EN-dec	$(2 + n)\|G_2\|$	——	$n\|G_1\| + (1 + 2n)\|G_2\|$	$(2 + 5n)\|G_2\|$
DU-dec	$3\|G_1\| + 3\|G_2\|$	$\|G_2\| + n\|Z_r\|$	$4\|G_1\| + 3\|G_2\| + 2\|Z_r\|$	$6\|G_2\| + 5\|Z_r\|$

Table 3 compares the computational costs of the proposed scheme to the schemes [20–22]. During system initialization, the computational cost of [21] fluctuates as the number of attributes increases, resulting in a higher computational overhead than other comparable schemes. Since the proposed scheme does not require exponential operations with relatively high computational overhead, when the number of user attributes is constant, the proposed scheme has the lowest computational overhead in the key generation phase compared to other schemes. The proposed scheme and [22] adopt outsourced encryption, outsourcing some encrypted computing to edge nodes, which reduces the computational burden of users. Since [22] uses more exponential operations, the computational overhead of this scheme in the encryption phase is lower. In the decryption phase, the proposed scheme, [21] and [22] outsourced a large number of decryption operations to edge nodes, while [20] did not outsource decryption. After partial decryption, the proposed scheme only performs multiplication operations, without pairing operations and exponential operations with high computational overhead. By comparison, the proposed scheme has the highest computational efficiency in the decryption phase.

The proposed scheme and alternative schemes are compared in Table 4 for communication overhead. In the system initialization, the communication overhead of the proposed scheme is the least, followed by [20], while [21] and [22] communication overheads all vary with the number of attributes. In the encryption phase, the proposed scheme and [22] use outsourcing technology, and part of the encryption is outsourced to edge nodes. By comparison, it can be seen that the users' communication overhead in this scheme is the lowest, followed by [20–22]. Suppose the communication overhead of edge nodes is added together. The communication overhead of our scheme on the group is less than that of [21]. Hence, the communication overhead of the scheme is minimal at this stage. In the decryption stage, the proposed scheme, [21] and [22] outsource part of the decryption to the edge nodes, dramatically reducing the user's computational burden. The storage burden of the vehicle is significantly reduced by using outsourced decryption, which is exceptionally critical for resource-constrained vehicle units in the IoV. Through comprehensive analysis, it can be concluded that the proposed scheme uses outsourcing technology to the edge nodes, which significantly improves the algorithm's efficiency and is more consistent with the actual Internet of Vehicles.

7.2. *Numerical analysis*

To obtain the computation time of the operations mentioned above, we simulated our scheme on a laptop's Linux system with an AMD Ryzen 7 4800 H 2.9 GHz CPU processor and a memory size of 8 GB. The PBC library was used to implement all algorithms.



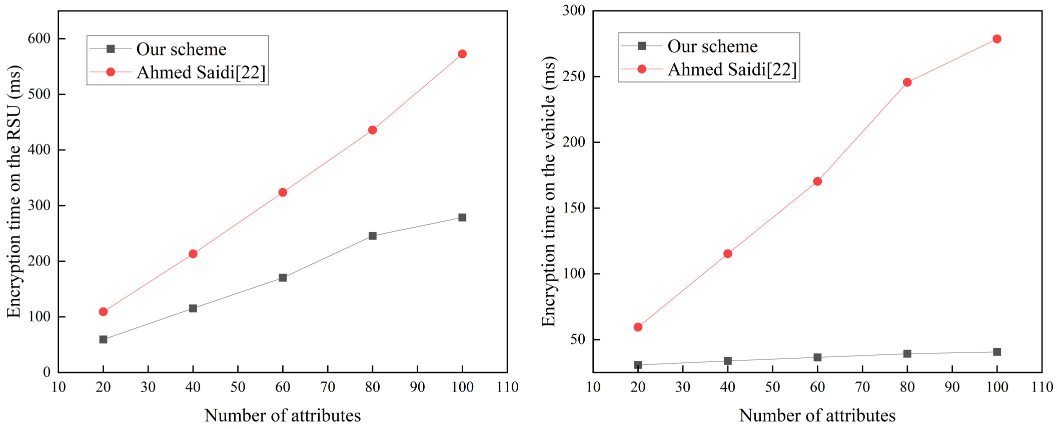
(a) Comparison of encryption time.

(b) Comparison of decryption time.

Figure 4. Comparison of encryption and decryption time.

By varying the amount of attributes, we test the time variation of the encryption algorithm, decryption algorithm and the entire algorithm in numerical simulations to demonstrate the effectiveness of our scheme. To simulate, we select attribute values of 20, 40, 60, 80 and 100. To minimize the error caused by the experimental environment factors, we ran the program 50 times and took the average value as its experimental result.

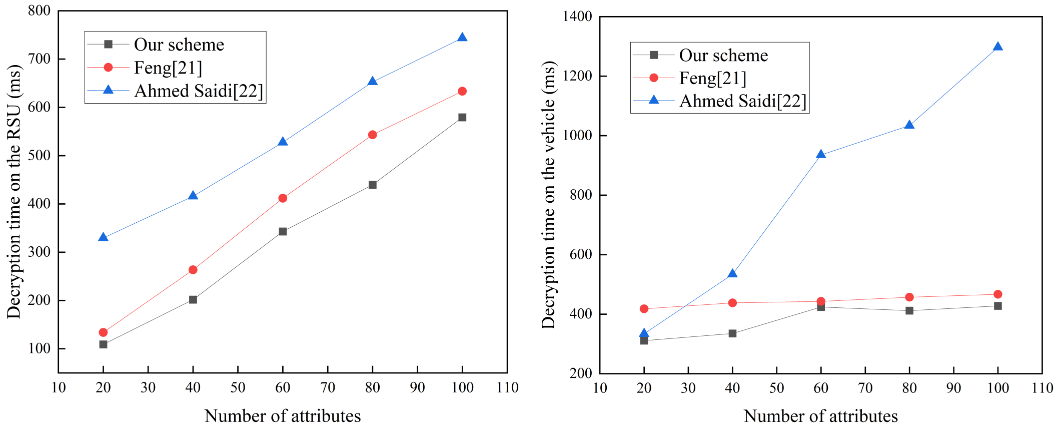
The experimental results of the computational overhead of the proposed scheme, Kang et al. [20], Feng et al. [21] and Saidi et al. [22] in the encryption stage and the decryption stage are shown in Figure 4. Compared with [20–22], with the increase of the number of attributes, our scheme needs less time to complete the whole encryption and decryption process and has higher computational efficiency.



(a) Comparison of encryption time on the RSU.

(b) Comparison of encryption time on the vehicle.

Figure 5. Comparison of encryption time on the RSU and the vehicle.



(a) Comparison of decryption time on the RSU.

(b) Comparison of decryption time on the vehicle.

Figure 6. Comparison of decryption time on the RSU and the vehicle.

The proposed scheme and [22] outsource part of the encryption operation to the edge node. Figure 5(a) represents the encryption time of the edge node, and Figure 5(b) represents the encryption time of the data owner. From Figure 5, we can find that the proposed scheme has lower computational overhead in both the outsourced encryption phase of the edge node and the outsourced decryption phase of the user, so our scheme has higher computational efficiency and lower latency.

In the decryption phase, the proposed scheme, [21] and [22] outsource part of the decryption operation to an edge node. Figure 6(a) shows that the scheme's edge node decryption is more efficient than the others. As shown in Figure 6(b), in the case of the same number of attributes, the user decryption of the proposed scheme takes the least time and is more efficient.

Eventually, we compared the overall running time of the schemes with the increase in the number of attributes, as shown in Figure 7. The proposed scheme is generally more efficient than [20–22]. Comprehensive analysis indicates that the scheme adopts outsourcing encryption and decryption, outsourcing some operations, which greatly improves the efficiency of the algorithm, reduces the computational burden of users, and is more suitable for the Internet of Vehicles environment with more resource-constrained devices.

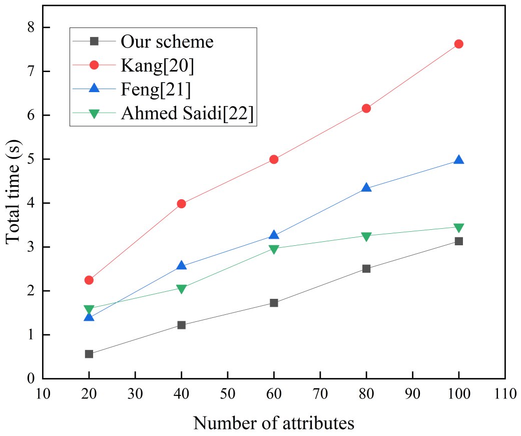


Figure 7. Computational costs of the total algorithm.

8. Conclusions

To enable safe communication between vehicles and infrastructure, a message-sharing scheme based on edge computing in the IoV is proposed. In the IoV, the selected location or specific type of vehicle is determined during message transmission. The attribute-based ciphertext policy approach introduced in this paper can provide message confidentiality and fine-grained access control. Additionally, due to the vehicular unit's limited computing and storage capacity in the IoV, we introduce edge computing and construct a three-layer architecture of the "cloud-edge-device", outsourcing some operations in the encryption and decryption to the RSU. Furthermore, the computational cost of vehicles is reduced and the computational efficiency of the scheme is improved dramatically. To protect the privacy of vehicle identity, we adopt an anonymous identity-based signature technology, and a trusted third party can trace the vehicle's real identity. At the same time, the proposal supports batch verification, reduces the computational burden of RSU verification messages and improves verification efficiency. Theoretical analysis and numerical results demonstrate that the proposed scheme has better advantages in computational efficiency than the existing schemes and is more suitable for practical IoV.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 62241207, 62262060), the Gansu Science and Technology Program (No. 22JR5RA158) and the Industrial support plan project of the Gansu Provincial Department of Education (No. 2022CYZC-17).

Conflict of interest

All authors declare there is no conflict of interest.

References

1. E. Sisinni, A. Saifullah, S. Han, U. Jennehag, M. Gidlund, Industrial internet of things: Challenges, opportunities, and directions, *IEEE Trans. Ind. Inf.*, **14** (2018), 4724–4734. https://doi.org/10.1109/TII.2018.2852491
2. F. Li, Y. Wang, Routing in vehicular ad hoc networks: A survey, *IEEE Veh. Technol. Mag.*, **2** (2007), 12–22. https://doi.org/10.1109/MVT.2007.912927
3. Y. Peng, Z. Abichar, J. M. Chang, Roadside-aided routing (rar) in vehicular networks, in *2006 IEEE International Conference on Communications*, **8** (2006), 3602–3607. https://doi.org/10.1109/ICC.2006.255631
4. L. Zhang, Q. Wu, J. Domingo-Ferrer, B. Qin, C. Hu, Distributed aggregate privacy-preserving authentication in vanets, *IEEE Trans. Intell. Transp. Syst.*, **18** (2017), 516–526. https://doi.org/10.1109/TITS.2016.2579162
5. V. Daza, J. Domingo-Ferrer, F. Sebe, A. Viejo, Trustworthy privacy-preserving car-generated announcements in vehicular ad hoc networks, *IEEE Trans. Veh. Technol.*, **58** (2009), 1876–1886. https://doi.org/10.1109/TVT.2008.2002581
6. F. Qu, Z. Wu, F. Y. Wang, W. Cho, A security and privacy review of vanets, *IEEE Trans. Intell. Transp. Syst.*, **16** (2015), 2985–2996. https://doi.org/10.1109/TITS.2015.2439292
7. S. S. Manvi, S. Tangade, A survey on authentication schemes in vanets for secured communication, *Veh. Commun.*, **9** (2017), 19–30. https://doi.org/10.1016/j.vehcom.2017.02.001
8. A. Wasef, X. Shen, Emap: Expedite message authentication protocol for vehicular ad hoc networks, *IEEE Trans. Mob. Comput.*, **12** (2013), 78–89. https://doi.org/10.1109/TMC.2011.246
9. P. Xu, S. He, W. Wang, W. Susilo, H. Jin, Lightweight searchable public-key encryption for cloud-assisted wireless sensor networks, *IEEE Trans. Ind. Inf.*, **14** (2018), 3712–3723. https://doi.org/10.1109/TII.2017.2784395
10. W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, et al., A survey on the edge computing for the internet of things, *IEEE Access*, **6** (2018), 6900–6919. https://doi.org/10.1109/ACCESS.2017.2778504

11. W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: Vision and challenges, *IEEE Internet Things J.*, **3** (2016), 637–646. https://doi.org/10.1109/JIOT.2016.2579198

12. P. Mach, Z. Becvar, Mobile edge computing: A survey on architecture and computation offloading, *IEEE Commun. Surv. Tutorials*, **19** (2017), 1628–1656. https://doi.org/10.1109/COMST.2017.2682318

13. I. A. Elgendy, W. Z. Zhang, H. He, B. B. Gupta, A. A. Abd El-Latif, Joint computation offloading and task caching for multi-user and multi-task mec systems: Reinforcement learning-based algorithms, *Wireless Netw.*, **27** (2021), 2023–2038. https://doi.org/10.1007/s11276-021-02554-w

14. Z. Kotulski, T. W. Nowak, M. Sepczuk, M. Tunia, R. Artych, K. Bocianiak, et al., Towards constructive approach to end-to-end slice isolation in 5g networks, *EURASIP J. Inf. Secur.*, **2018** (2018), 1–23. https://doi.org/10.1186/s13635-018-0072-0

15. X. Foukas, G. Patounas, A. Elmokashfi, M. K. Marina, Network slicing in 5g: Survey and challenges, *IEEE Commun. Mag.*, **55** (2017), 94–100. https://doi.org/10.1109/MCOM.2017.1600951

16. J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption, in *2007 IEEE Symposium on Security and Privacy (SP'07)*, (2007), 321–334. https://doi.org/10.1109/SP.2007.11

17. J. Cheng, G. Yuan, M. Zhou, S. Gao, C. Liu, H. Duan, A fluid mechanics-based data flow model to estimate vanet capacity, *IEEE Trans. Intell. Transp. Syst.*, **21** (2020), 2603–2614. https://doi.org/10.1109/TITS.2019.2921074

18. D. Huang, M. Verma, Aspe: Attribute-based secure policy enforcement in vehicular ad hoc networks, *Ad Hoc Networks*, **7** (2009), 1526–1535. https://doi.org/10.1016/j.adhoc.2009.04.011

19. H. Cui, R. H. Deng, G. Wang, An attribute-based framework for secure communications in vehicular ad hoc networks, *IEEE/ACM Trans. Networking*, **27** (2019), 721–733. https://doi.org/10.1109/TNET.2019.2894625

20. Q. Kang, X. Liu, Y. Yao, Z. Wang, Y. Li, Efficient authentication and access control of message dissemination over vehicular ad hoc network, *Neurocomputing*, **181** (2016), 132–138. https://doi.org/10.1016/j.neucom.2015.06.098

21. C. Feng, K. Yu, M. Aloqaily, M. Alazab, Z. Lv, S. Mumtaz, Attribute-based encryption with parallel outsourced decryption for edge intelligent IoV, *IEEE Trans. Veh. Technol.*, **69** (2020), 13784–13795. https://doi.org/10.1109/TVT.2020.3027568

22. A. Saidi, O. Nouali, A. Amira, Share-abe: An efficient and secure data sharing framework based on ciphertext-policy attribute-based encryption and fog computing, *Cluster Comput.*, **25** (2022), 167–185. https://doi.org/10.1007/s10586-021-03382-5

23. X. Liu, Z. Shan, L. Zhang, W. Ye, R. Yan, An efficient message access quality model in vehicular communication networks, *Signal Process.*, **120** (2016), 682–690. https://doi.org/10.1016/j.sigpro.2014.11.012

24. T. Chim, S. Yiu, L. C. Hui, V. O. Li, Vspn: Vanet-based secure and privacy-preserving navigation, *IEEE Trans. Comput.*, **63** (2014), 510–524. https://doi.org/10.1109/TC.2012.188

25. Y. Jiang, Y. Ji, T. Liu, An anonymous communication scheme based on ring signature in vanets, preprint, arXiv:1410.1639.

26. M. Verma, D. Huang, Segcom: Secure group communication in vanets, in *2009 6th IEEE Consumer Communications and Networking Conference*, (2009), 1–5. https://doi.org/10.1109/CCNC.2009.4784943

27. L. Zhang, Q. Wu, A. Solanas, J. Domingo-Ferrer, A scalable robust authentication protocol for secure vehicular communications, *IEEE Trans. Veh. Technol.*, **59** (2010), 1606–1617. https://doi.org/10.1109/TVT.2009.2038222

28. K. A. Shim, $\mathcal{CPAS}$: An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks, *IEEE Trans. Veh. Technol.*, **61** (2012), 1874–1883. https://doi.org/10.1109/TVT.2012.2186992

29. J. C. Choon, J. Hee Cheon, An identity-based signature from gap diffie-hellman groups, in *Public Key Cryptography — PKC 2003*, (2002), 18–30. https://doi.org/10.1007/3-540-36288-6_2

30. S. Niu, H. Shao, Y. Hu, S. Zhou, C. Wang, Privacy-preserving mutual heterogeneous signcryption schemes based on 5g network slicing, *IEEE Internet Things J.*, **9** (2022), 19086–19100. https://doi.org/10.1109/JIOT.2022.3163607