Pan et al. EURASIP Journal on Wireless Communications and Networking (2019) 2019:169
https://doi.org/10.1186/s13638-019-1494-1

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Secure data sharing scheme for VANETs based on edge computing

Jingwen Pan, Jie Cui*, Lu Wei, Yan Xu and Hong Zhong

Abstract

The development of information technology and the abundance of problems related to vehicular traffic have led to extensive studies on vehicular ad hoc networks (VANETs) to meet various aspects of vehicles, including safety, efficiency, management, and entertainment. In addition to the security applications provided by VANETs, vehicles can take advantage of other services and users who have subscribed to multiple services can migrate between different wireless network areas. Traditionally, roadside units (RSUs) have been used by vehicles to enjoy cross-domain services. This results in significant delays and large loads on the RSUs. To solve these problems, this paper introduces a scheme to share data among different domains. First, a few vehicles called edge computing vehicles (ECVs) are selected to act as edge computing nodes in accordance with the concept of edge computing. Next, the data to be shared are forwarded by the ECVs to the vehicle that has requested the service. This method results in low latency and load on the RSUs. Meanwhile, ciphertext-policy attribute-based encryption and elliptic curve cryptography are used to ensure the confidentiality of the information.

Keywords: VANETs, edge computing, data sharing, ECC, ABE, cross-domain service

1 Introduction

Vehicular ad hoc networks (VANETs) is an application of the mobile ad hoc network in the transportation field and is a multi-hop mobile wireless communication network that was first mentioned in 2001 [1]. The core idea of VANETs is that vehicles are automatically connected to the mobile network within a specific communication range and these connected vehicles are able to exchange information, such as speed, location, and data sensed by on-board sensors. VANETs communication modes include inter-vehicle communication (V2V, vehicle-to-vehicle) and communication with public infrastructure (V2I, vehicle-to-infrastructure) that realize real-time information exchange and serve people's transportation [2].

VANETs typically consist of three parts, namely, a trusted authority (TA), a roadside unit (RSU), and an on-board unit (OBU). The TA, which is a trusted management center with high computing capacity and storage, is responsible for registering and issuing secret key materials. The RSU, located on both sides of the roads, interacts with vehicles through wireless channels and serves as a bridge between the vehicles and the TA. OBU, which is a computing device equipped in a vehicle, is in charge of V2V and V2I communications, dealing with the release and reception of traffic messages and improving the user's driving experience [3].

Through RSUs, VANETs can provide convenient services such as notifying the nearest restaurant and gasoline stations and acting as a gateway by connecting with the Internet and mobile communication network to provide electronic toll collection service and in-vehicle entertainment services, such as downloading movies. Users can effectively monitor the driving condition of vehicles with respect to different functional requirements and provide comprehensive services that can greatly facilitate passengers' travels and enrich their travel journeys.

Development of information technology has given rise to the demand for information gathering and information sharing among different networks. Data collected within one domain may not satisfy users because the required data may be present in another management domain. For example, a Twitter user may want to share data with another user who has an account on Instagram but not on Twitter. Therefore, a secure way to share data between different domains is needed. The main

* Correspondence: cuijie@mail.ustc.edu.cn
School of Computer Science and Technology, Anhui University, Hefei 230039, Anhui, China

problems encountered during cross-domain sharing are data security and authentication among different domains [4]. The dynamic movement of vehicle nodes in VANETs has given rise to the authentication problem during cross-domain access. The traditional method for vehicles to enjoy cross-domain services relies on RSUs and causes significant delays and large loads on the RSUs.

We have introduced the concept of edge computing to overcome the cross-domain sharing problem. Edge computing is a technology that allows computing to be performed at the edge of a network so that computing occurs near data sources [5]. Traditionally, transmission, storage, and computing have been designed separately for the convenience of management; however, these separate resources cannot satisfy the latency and quality requirements of the service. By contrast, edge computing allows deep integration of transmission, storage, and computing. For example, because edge computing allocates a large amount of computing power near a mobile device, such as a vehicle, the majority of the data are processed and stored at the edge, thereby decreasing the delay in providing information and computing resources to the user. In addition, at present, the shared pools for configuring various resources (e.g., computing networks, servers, storage, applications and services) are centrally located to facilitate management, coherence, and economy. However, a fully centralized approach impedes large-scale connections, large-scale transmissions, and latency. Therefore, edge computing is used to realize the decentralization of shared resources that are distributed along the continuum from the cloud to things. Centralization improves efficiency and flexibility, while decentralization decreases latency and improves capacity and scalability [6, 7].

The resources of computing, communication, storage, and control are distributed among all the nodes. The structure of nodes along with their capabilities of storage, computing, and networking are different. It is imperative that the various communication modes as well as the nodes cooperate with each other to optimize the use of resources and improve the performance of data sharing. Furthermore, a few edge computing nodes are needed to function as data sources. An edge computing node can be perceived as the administrator of one domain and is responsible for sending and processing data [5]. Edge computing nodes can reduce delays by processing and analyzing simple data generated by edge devices, passing on only the necessary results and complex data to the remote cloud. By aggregating information from edge computing nodes, a cloud service provider (CSP) can obtain real-time traffic and data requests and, subsequently, schedule data caching and coordinate resources among different domains. In our research, we select several vehicles to act as edge computing nodes. The points to be considered when selecting an edge computing node are as follows: (1) availability of adequate computing resources, (2) high social centrality implying a greater possibility to contact other nodes and a high data sharing efficiency, and (3) availability and selection of vehicles that offer a wider coverage of routes to share data among a large number of nodes. All OBUs communicate with edge computing nodes and provide information that includes the list of their current neighbors, the channel capacity of each neighbor's link, and the identifiers of the cached and un-cached data items. Subsequently, edge computing nodes inform the published scheduling decisions to all the relevant OBUs. Next, each node obtains the shared data that they requested from their neighbors based on the scheduling decisions [3, 6].

In this paper, we propose a secure scheme based on edge computing for data sharing among different domains. As shown in Fig. 1, in accordance with the concept of edge computing, we select some vehicles, called edge computing vehicles (ECVs), to act as edge computing nodes. ECVs integrate the information obtained from the OBUs and the RSUs and schedule the data conforming to the requests made by vehicles. Each ECV manages the domain it resides in. Requests and responses for data sharing between two domains are transmitted through the CSP and their respective ECVs. This shared data is encrypted by elliptic curve cryptography (ECC). The message that needs to be shared among users in two domains uses the resource pool as its carrier for access and storage and is encrypted by ciphertext-policy attribute-based encryption (CP-ABE) to ensure confidentiality. Different from the traditional public key encryption, the attribute-based encryption (ABE) algorithm embeds the attribute set and policy into the ciphertext and user's private key so that the decryption process is actually matching the set of attributes with the strategy. If the matching is successful, the algorithm will complete the decryption operation and the user will recover the plaintext data. CP-ABE, with policy being embedded in ciphertext, which means that the data owner can determine those who have access to the ciphertext by setting policies, makes an encryption access control for the data that can refine the granularity to the attribute level. Therefore, from the perspective of encryption calculation overhead and storage overhead, CP-ABE has an advantage in performance compared to the traditional public key encryption algorithm in the data encryption sharing scenario. Throughout this process, we use system parameters that are set by TA and stored in a tamper-proof device (TPD) to perform pseudo-identity-based message signing and authentication that guarantees the anonymity of the message owner and security of the message transmission.

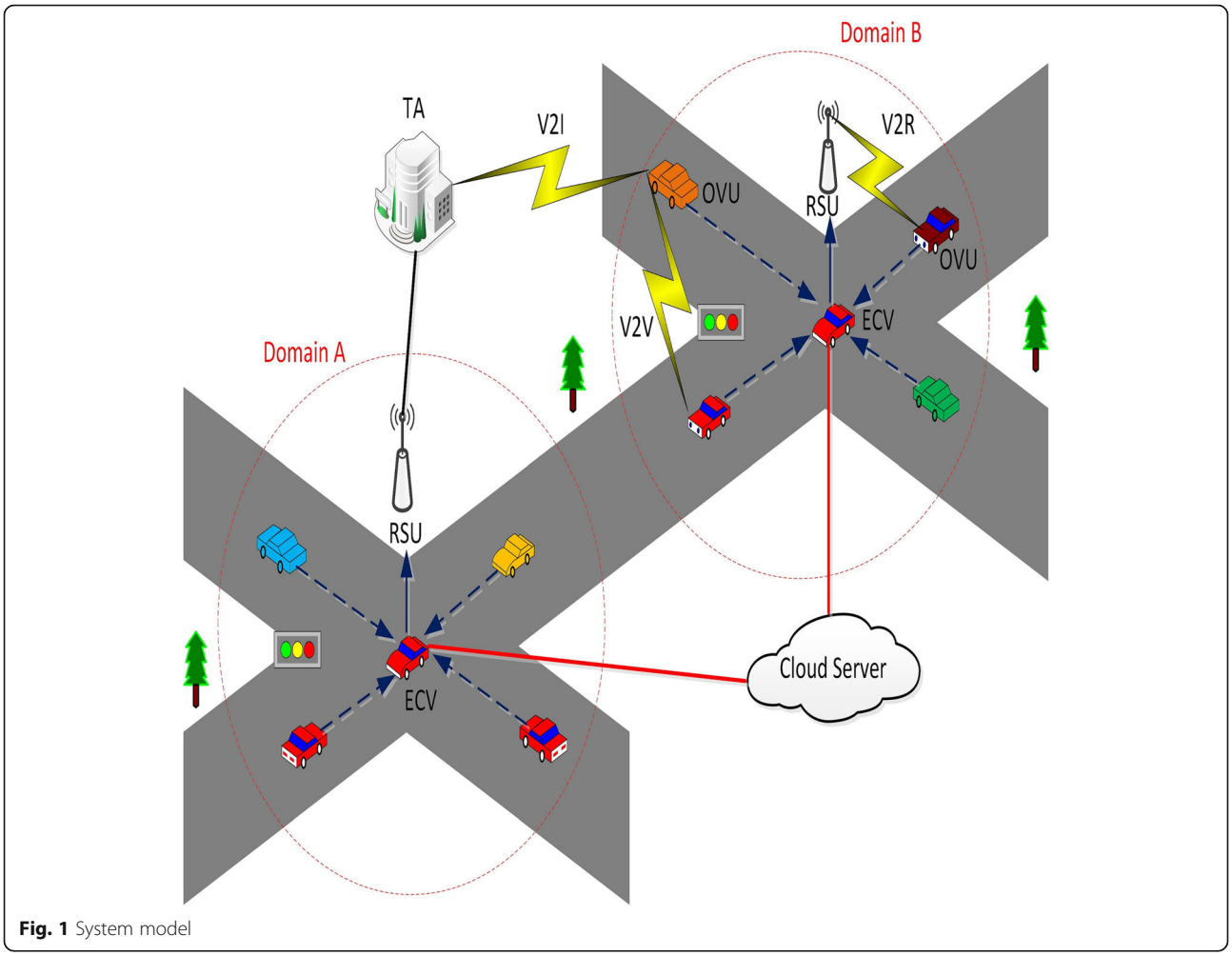

Fig. 1 System model

Our solution enables efficient and dynamic data sharing between vehicles in different domains, which will greatly facilitate vehicle users in practical applications. For example, there is a moving vehicle with a lower oil level that needs to know the location of nearby gas stations, then by using our scheme, it can quickly obtain real-time information through the interaction of ECVs. As far as we know, there is no research on combining ECC with edge computing for cross-domain data sharing, so we explored and proved its feasibility and significance.

The main contributions of this paper can be summarized as follows:

(1) We propose a scheme that uses edge computing to achieve cross-domain sharing, which can improve the efficiency of vehicles to obtain the data they need for service and greatly reduce the load on RSUs.
(2) We apply a method that combines ECC, CP-ABE, and a message signature authentication mechanism and provide a security analysis that proves the security of our scheme.

The rest of this paper is organized as follows: Section 2 briefly mentions the related work. In Section 3, we introduce the fundamental background of ECC and CP-ABE. Thereafter, our proposed scheme is described in Section 4. Proof and analysis are provided in Section 5. Finally, we conclude our work and discuss future research directions in Section 6.

As for the experiment in our paper, we conduct an experiment to evaluate the performance of the proposed solution. Our experiments are tested on Ubuntu 14.04 platform with two cryptography libraries including MIRACL [8] and Crypto++ [9]. By measuring the computation time of some basic cryptography operations, we learn about the computation performance of our proposed scheme and compare it with other related schemes. The experimental results show that compared with the other two existing data sharing schemes, our scheme can reduce the computation overhead in the process of transmitting messages which means that our proposed scheme can suit the real VANETs scenario better. Besides, the cryptography and correctness analysis

are used to guarantee that our proposed can resist common attacks of VANETs.

2 Related work

2.1 Privacy protection of VANETs

In 2006, Zeng [10] proposed a pseudonym public key infrastructure (PKI) solution based on public key infrastructure, in which vehicles can generate pseudonyms themselves so as to reduce the overhead of communication with the certification authority (CA). In 2007, Lin et al. [11] presented a privacy protection protocol based on the combination of a group signature and an identity-based signature (IBS). Anonymity and traceability can be guaranteed by using a short group signature to sign messages, while bandwidth can be saved by using the identity-based signature scheme. However, the group signature schemes have such problems as maintenance of the revocation list. In 2008, Zhang et al. [12] introduced an efficient batch signature verification scheme, intending to solve the problem that it is difficult for RSUs to simultaneously verify multiple received signatures in V2I communication mode. The scheme can greatly reduce the overall time and transmission overhead, but it is vulnerable to replays and non-repudiation attacks. In 2015, Horng et al. [13] proposed a scheme based on the certificateless signature, which solved the complex certificate management problem of the traditional PKI-based schemes and the key escrow problem in IBS. Conditional privacy protection will be achieved by mapping the message broadcast by vehicles to different pseudo-identities. And authorities can retrieve real identity from any controversial pseudo-identity. However, this scheme only considers V2I communication and lacks support for malicious vehicles revocation. In 2016, Vijayakumar et al. [14] proposed a dual authentication scheme and a dual group key management scheme, which have high computational efficiency and can safely distribute group keys to vehicle groups. However, it is vulnerable to replay attacks when reusing previously acquired messages. In 2017, Azees et al. [15] presented an efficient anonymous authentication scheme which has an efficient tracking method to avoid malicious vehicles entering VENETs. But the scheme leaves out of considering non-framework, which can guarantee that members' signatures are not forged by others. A new efficient certificateless short signature (CLSS) scheme is designed by Tsai [16] using bilinear pairing, which takes a group element as the signature length and takes on the lower computational cost of signature generation and signature verification. After a formal security analysis, the solution proposed proved to be safe for both super I and super II opponents. In 2018, Pournaghi et al. [17] proposed a scheme based on a combination of RSUs and TPD. Storing the system key and main parameters in the TPD of RSUs ensures that the entire network will not be affected too much when a single OBU hazard or attack occurs. Asaar et al. [18] found that the authentication scheme proposed by Liu et al. [19] using proxy vehicles to reduce the computational overhead of RSUs does not guarantee the authenticity of the message, nor can it resist the modification of the attack and the invalid signature of the batch. So, they designed a new identity-based message authentication scheme using proxy vehicles and demonstrated the security of the scheme on the elliptic curve discrete logarithm problem. Islam et al. [20] introduced a password-based conditional privacy protection authentication and group key generation (PW-CPPA-GKA) protocol for VANETs, which can provide some functions like group-key generation, user departing, user joining, and password modification. Because PW-CPPA-GKA is bilinear-pairing-free, it is lightweight in terms of computation and communication. Cui et al. [21] proposed an authentication scheme using the Cuckoo filter. In their scheme, the ideal TPD is no more necessary and the computation overhead is very low.

In recent years, researchers have made great progress in the privacy protection of vehicle network [22–24].

2.2 Data downloading or sharing

In 2007, Sago et al. [25] grouped vehicles according to locations and estimated future routes. Then, they made predictions about the data items that might be transmitted between different groups in the near future in order to improve the availability of data shared between vehicles. In 2010, Zhang et al. [26] intended to improve the performance of content sharing in VANETs and proposed Roadcast, a popularity-based P2P sharing scheme. On the one hand, Roadcast relaxes the query requirements of users and makes it faster for users to query the content they want. On the other hand, Roadcast returns the most popular content related to queries under the influence of two components (popularity aware content retrieval and popularity aware data replacement) and increases opportunities for spread and share of popular data. Therefore, the overall query delay is reduced. However, data transmitted between any two parties may get compromised, and several attack method has been proposed such as [27]. Hence, data privacy and security issue should be paid attention to. Some efficient schemes that focus on solving these issues have been proposed such as [28–30]. In 2013, Hao et al. [31] proposed a secure co-downloading framework for paid services of VANETs. Data downloading takes place when the vehicles enter the range of RSUs and data sharing takes place after the vehicles leaving it. The application layer data sharing protocol they proposed coordinates the vehicles based on location to transfer the data to be

shared. This cooperative sharing can effectively avoid conflicts in the media access control (MAC) layer and hidden terminal problems in multi-hop transmission and can ensure that each vehicle near the RSUs can receive the requested data. In 2014, Wu et al. [32] used evolutionary games (EG) to implement multimedia services and data sharing among VANETs vehicles. This scheme presents a repeated game "More Pay for More Work (RGMPMW)" incentive mechanism based on service evaluation information. In 2017, Lai et al. [33] proposed an effective cloud-assisted scheme for data storage and query in VANETs. The cloud calculates the transfer strategy of the data query result by solving the linear programming problem. This scheme integrates the cloud, in-vehicle network, and 4G technology, and processes and transfers queries to corresponding communication channels based on the cost and time of the query, which greatly improves efficiency.

2.3 Edge computing

Cloud computing service which mainly contains SaaS (software as a service), LaaS (infrastructure as a service), PaaS (platform as a service) [34] is very popular in recent years because it can decrease the terminal running costs. However, cloud computing cannot process data timely. Given the recent proliferation in the number of smart devices connected to the Internet, the era of Internet of Things (IoT) is challenged with massive amounts of data generation. Edge computing or fog computing is gaining popularity and is being increasingly deployed in various latency-sensitive application domains including industrial IoT [35].

In 2016, Shi et al. [36, 37] described the application prospects of edge computing, pointing out that edge computing will play an important role in solving delays, limited battery life, bandwidth cost, data security, and privacy issues. In 2017, Mao et al. [38] conducted a comprehensive survey of MEC from the perspective of communication, discussed some challenges and directions of the research, including MEC system deployment, mobility management, and privacy awareness, and introduced some typical application scenarios of edge computing. Ren et al. [39] studied the application of edge computing in the field of Internet of Things. They implemented an extensible Internet of Things platform based on transparent computing using edge computing, which proves that edge computing can enhance the scalability of lightweight Internet of Things devices. In 2018, Roman et al. [40] introduced several of the most important edge examples, which indicated the challenges and potential synergies of mobile edge computing (MEC). Yuan et al. [41] studied how to meet the need of real-time access services in the autonomous driving process using MEC technology and proposed a two-level edge computing architecture to coordinate vehicular content sharing by making full use of base stations on wireless edge. The simulation results show that the proposed solution can significantly reduce the backhaul and wireless bottleneck of the cellular network.

Fan et al. [5] first linked edge computing to cross-domain access. The proposed edge computing model effectively solves the authentication problem between different domains through edge computing nodes and cloud links. Meanwhile, the RSA algorithm and CP-ABE guarantee scheme security to achieve cross-domain sharing. Luo et al. [6] studied the distribution problem of vehicular content in 5G-VANETs. In order to allocate large amounts of data, a two-layer hierarchical structure based on edge computing was designed. The upper layer coordinates base station resources and handles unbalanced traffic, while the macro base station (MBS) of the lower layer supports cooperation among different communications and coordinates content requests among vehicles. After data prefetching into RSUs and vehicles, the RSUs and the vehicles act as data sources to provide content download services for the neighboring vehicles, and the data is scheduled by the MBS and propagated between RSUs and the vehicles.

3 Background

3.1 Elliptic curve cryptography

ECC is an algorithm based on elliptic curve mathematics for public key cryptography, which is first proposed by Miller [42] and Koblitz [43]. Under the same security conditions, ECC has a key with a shorter length compared to other public key cryptographic algorithms. An elliptic curve is a collection of points that satisfy a particular equation, while a definite elliptic curve cryptosystem can be determined by a maximum prime number, an elliptic curve equation, and a common point on the curve.

For an elliptic curve equation E, there are an infinity point O and an operator + called addition in the mathematical principle. It has the following properties:

(1) Unit element: $P + O = O + P = P$, for all $P \in E$.
(2) Reversibility: $P + (-P) = O$, for all $P \in E$.
(3) Associative law: $(P + Q) + R = P + (Q + R)$, for all P, Q, $R \in E$.
(4) Commutative law: $P + Q = Q + P$, for all P, $Q \in E$.
(5) Specific calculation: Given two points P_1 and P_2 on E, there must be a third point $P_3 = P_1 + P_2$ on E, and it can be determined by the connection between P_1 and P_2.
(6) Multiplication: Ellipse scale multiplication is an extension of elliptical addition. Given the point P on E, then $kP = P + P + \ldots + P(k \text{ times})$.

In ECC, given the elliptic curve E, the base point G , and the point xG, then we take xG as the public key and take x as the private key. According to the natures of the elliptic curve, we can know that it is very simple to obtain the public key when the private key is known, but it is quite hard to find the private key when the public key is known. This is the elliptic curve discrete logarithm problem (ECDLP), whose difficulty guarantees the security of the elliptic curve cryptography.

3.2 Ciphertext-policy attribute-based encryption

In 2017, Bethencourt et al. [44] proposed the CP-ABE scheme. In CP-ABE, the ciphertext corresponds to the access structure while the key corresponds to the set of attributes. The encryption part encrypts data using public parameters and the user decrypts the ciphertext using the attribute-based private key. Since the policy is embedded in the ciphertext, the data owner can define access control policy to determine users who can access the ciphertext. The process of CP-ABE is as follows:

(1) Setup: The setup algorithm outputs the public parameter PK and a master key MK.
(2) Encryption: The encryption algorithm inputs a message m and an access structure A and PK, then outputs the ciphertext C.
(3) Generate key: The algorithm inputs a set of attributes S, MK, and PK, and outputs a decryption key D.
(4) Decryption: The decryption algorithm inputs the key D, the public parameter PK, and the ciphertext C encrypted based on the access structure A. If the number of attributes that can satisfy A in all the attributes corresponding to the user D reaches a certain threshold, then the user can decrypt and gain the message m.

4 Our scheme

4.1 System model

As shown in Fig. 1, our model consists of five types of entities: a trusted authority (TA), a roadside unit (RSU), a cloud service provider (CSP), edge computing vehicle (ECV), and ordinary vehicle user (OVU).

(1) TA: As a trusted management center with high computing capacity and storage, TA generates and publishes common system parameters about the secret key to all vehicles.
(2) RSU: Besides serving as a bridge between the vehicles and the TA, RSUs also provide the information obtained from vehicles to the ECV and participate in the transferring of data under the control of ECV.
(3) CSP: It links all ECVs so that those domains managed by ECVs can have contact with each other and the sharing of data can be implemented between those different domains.
(4) ECV: An ECV, which is responsible for transmission and storage of data as well as users' registration and revocation, manages a domain. After receiving the request of data sharing from another domain, the ECV encrypts the list of attributes of its domain with ECC.
(5) OVU: OVUs can either encrypt data according to the policy and send it to the resource pool as data requesters or can access and decrypt the data of the resource pool as other users.

4.2 System initialization phase

We need some necessary system parameters which are generated by the TA and preloaded into the tamper-proof device (TPD) of all vehicles.

TA randomly selects two large prime numbers p and q, a non-singular elliptic curve$E: y^2 = x^3 + ax + b \bmod q$, and a generator element G randomly selected in the group.

TA randomly selects $k_s \in Z_q^*$ as the system private key and calculates $K_s = k_s G$ as the system public key.

TA randomly selects $k_R \in Z_q^*$ as the private key of RSUs, calculates $K_R = k_R G$ as the public key of RSUs, and sends k_R to RSUs.

TA chooses a secure hash function: $h: \{0,1\}^* \rightarrow Z_q$.

TA assigns a real identity RID and password PWD to each vehicle and preloads $\{\mathrm{RID}, \mathrm{PWD}, k_s\}$ into the TPD of the vehicle.

TA randomly selects two numbers α, $\beta \in Z_q$ as public parameters for encryption and decryption.

TA publishes common system parameters $\{p, q, a, b, G, K_s, K_R, h, \alpha, \beta\}$ to all vehicles.

4.3 ECV election method

In this paper, we consider two criteria when selecting ECVs: closer distance from RSUs and enough available computing resources. In our proposed scheme, we adopt the same election method as that of the scheme of Cui et al. [3].

4.4 Process of constructions

Assuming that the OVU in domain A denoted by $\mathrm{OVU_A}$ wants to share data with users in domains B, first, $\mathrm{OVU_A}$ sends data sharing request and public key K_A to the edge computing vehicle $\mathrm{ECV_B}$ in domain B through $\mathrm{ECV_A}$ and CSP. Then, $\mathrm{ECV_B}$ derives the symmetric encryption key S from K_A and returns the attributes set of domain B which encrypted by S to $\mathrm{OVU_A}$ through CSP and $\mathrm{ECV_B}$. Finally, $\mathrm{OVU_A}$ defines policy according to the set of attributes and encrypts the message as well as sends it to the resource pool where users in domain B can access to. Here, we describe the specific process of the proposed scheme in detail, as shown in Fig. 2.

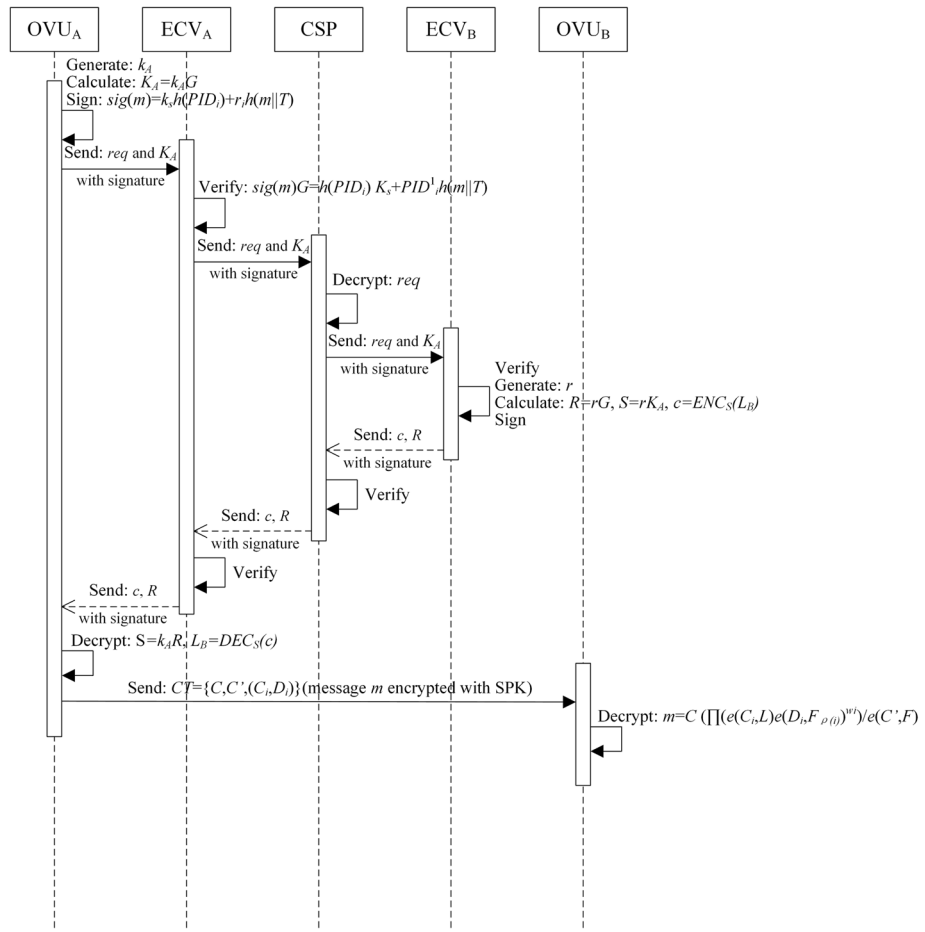

Fig. 2 Operations and interaction process of proposed scheme

(1) Requests by $\mathrm{OVU_A}$: $\mathrm{OVU_A}$ randomly generates $k_A \in [1, n-1]$ as the private key and then calculates $K_A = k_A G$ as the public key. And req denotes the request of sharing data with users who are in domain B. Before sending a message, $\mathrm{OVU_A}$ must complete the following work so that K_A and req can be sent securely to $\mathrm{ECV_A}$.

First, $\mathrm{OVU_A}$ needs to send its real identity RID and password PWD to TPD for authentication. If these two values are inconsistent with the pre-stored values in the TPD, the authentication will fail and the next service will be rejected. After the identity is successfully verified, TPD will calculate the pseudo-identity $\mathrm{PID}_i = \{\mathrm{PID}_i^1, \mathrm{PID}_i^2\}$, where i denotes the number of the vehicle, r_i is a random number generated by TPD, $\mathrm{PID}_i^1 = r_i \cdot G$, $\mathrm{PID}_i^2 = \mathrm{RID} \oplus h(r_i \cdot K_s)$. Besides, in order to prevent messages from being tampered with during transmission, $\mathrm{OVU_A}$ must provide signatures for all messages to be sent. To send the message m, the signature function is defined as $\mathrm{sig}(m) = k_s h(\mathrm{PID}_i) + r_i h(m \| T)$, where T is the current timestamp. So, the content sent by the vehicle is such a message signature pair $\{\mathrm{PID}_i, T, m, \mathrm{sig}(m)\}$.

Therefore, $\mathrm{OVU_A}$ sends $\{\mathrm{PID}_{\mathrm{OVU_A}}, T_{\mathrm{OVU_A}}, K_A, \mathrm{req}, \mathrm{sig}(K_A), \mathrm{sig}(\mathrm{req})\}$ to $\mathrm{ECV_A}$ finally.

(2) $\mathrm{ECV_A}$ and CSP process: After receiving the message, $\mathrm{ECV_A}$ verifies the integrity of the message and the legality of the message signature. According to some formulas above, we can know that:

$$\begin{aligned}\mathrm{sig}(m) \cdot G &= G \cdot k_s h(\mathrm{PID}_i) + G \cdot r_i h(m \| T) \\ &= K_s \cdot h(\mathrm{PID}_i) + \mathrm{PID}_i^1 \cdot h(m \| T)\end{aligned}$$

Thus the equation $\mathrm{sig}(m) \cdot G = K_s \cdot h(\mathrm{PID}_i) + \mathrm{PID}_i^1 \cdot h(m \| T)$ can be used to verify messages. If the calculation results on the left are equal to the one on the right, the verification is successful. Otherwise, the verification fails.

$\mathrm{ECV_A}$ sends the message to CSP. After receiving the message, CSP decrypts the request to determine that the

domain OVU_A wants to share with is domain B, and then forwards the message to ECV_B.

(3) ECV_B return: After verifying the public key K_A and the request for data sharing *req* from domain A, ECV_B encrypts the attribute list of domain B with symmetrical encryption to return to domain A.

ECV_B generates a random number $r \in [1, n-1]$ and calculates the intermediate parameter $R = rG$ and symmetric key $S = rK_A$.

After that, ECV_B uses S and symmetric encryption scheme to encrypt the attribute list of domain B denoted byL_B, and the encrypted ciphertext is $c = \mathrm{ENC}_S(L_B)$.

Finally, ECV_B sends the message signature pair $\{\mathrm{PID}_{\mathrm{ECV_B}}, T_{\mathrm{ECV_B}}, R, c, \mathrm{sig}(R), \mathrm{sig}(c)\}$ signed for R and c to OVU_A through CSP and ECV_A.

(4) Decryption by OVU_A: After OVU_A successfully verifies the message received, decryption is needed to obtain L_B.

First, according to the known conditions, the following equation exists: $S = rK_A = rk_AG = k_AR$. Then, we can calculate $S = k_AR$ to get the same key S as generated by ECV_B and take it as the session key. Finally, the attributes list L_B can be gained by decryption with the symmetric encryption scheme: $L_B = \mathrm{DEC}_S(c)$.

(5) Encryption by OVU_A: Based on the attribute list, OVU_A can use CP-ABE scheme to encrypt data and define a policy to determine users who can access the ciphertext.

We define system public key as SPK = g, $e(g,g)^{\alpha}$, g^{β}, f_1, … , f_X, where g is a generator, random numbers $\alpha, \beta \in Z_q$, and random numbers $f_1, \ldots, f_X$ correspond to the x attributes of L_B.

OVU_A defines access control policy (M, ρ). M is a share-generating matrix with x rows and y columns. For $i = 1, \ldots, x$, function $\rho(i)$ associates the ith row of matrix M to an attribute of list L_B. OVU_A selects a random vector $\overrightarrow{v} = (\gamma, t_2, \ldots, t_y) \in Z_q^y$, where $t_2, \ldots, t_y$ are randomly chosen to share γ, then calculates $\lambda_i = \overrightarrow{v} \cdot M_i$, where M_i is the vector corresponding to the ith row of M. $\{\lambda_i = (M\overrightarrow{v})_i\}_{i \in \{1,\ldots,x\}}$ are valid shares only when there is a set of constants $\{w_i \in Z_q\}_{i \in \{1, \ldots, x\}}$ such that the equation $\sum_{i \in \{1,\ldots,x\}} w_i \lambda_i = \gamma$ holds. In this case, the user can decrypt the ciphertext.

Assuming that OVU_A wants to share the message m, the system public key SPK is used to encrypt m. First, calculate $C = me(g,g)^{\alpha\gamma}$ and $C' = g^{\gamma}$. Meanwhile, for all rows of the matrix M, i.e., for $i = 1, \ldots, x$, calculate $C_i = g^{\beta\lambda_i} f_{\rho(i)}^{-r_i}$ and $D_i = g^{r_i}$, where $r_1, \ldots, r_x \in Z_q$ are chosen randomly by OVU_A.

Therefore, the ciphertext that published by OVU_A is $CT = \{C, C', (C_i, D_i)\}_{i \in \{1, \ldots, x\}}$.

Finally, OVU_A sends CT to the public resource pool to which the users in domain B can access.

(6) Users decryption: Users use attribute-based privacy key to decrypt the ciphertext. The private key of a user with attributes A in domain B is defined as:

$$F = g^{\alpha} g^{\beta\varepsilon}, L = g^{\varepsilon}, \forall i \in A : F_i = f_i^{\varepsilon}$$

Users whose attributes satisfy the access structure can gain the message m by calculating the following formula:

$$m = C \cdot \left(\Pi_{i \in \{1,\ldots,x\}} \left(e(C_i, L) e\left(D_i, F_{\rho(i)}\right)\right)^{w_i}\right) / e(C', F)$$

5 Analysis of our scheme

In this section, the correctness proof and security analysis and efficiency analysis of our scheme are given.

5.1 Correctness of the CP-ABE scheme

A user who is qualified to access and decrypt the ciphertext has attributes that satisfy the access structure, which means his $\{\lambda_i = (M\overrightarrow{v})_i\}_{i \in \{1,\ldots,x\}}$ are valid. Thus, there exist constants $\{w_i \in Z_q\}_{i \in \{1, \ldots, x\}}$ to make the equation $\sum_{i \in \{1,\ldots,x\}} w_i \lambda_i = \gamma$ set up. The correctness of the decryption algorithm is proved as follows:

$$\frac{C \cdot \left(\Pi_{i \in \{1,\ldots,x\}} \left(e(C_i, L) e\left(D_i, F_{\rho(i)}\right)\right)^{w_i}\right)}{e(C', F)} =$$

$$\frac{C \cdot \left(\Pi_{i \in \{1,\ldots,x\}} e(g,g)^{\varepsilon\beta\lambda_i w_i}\right)}{e(g,g)^{\alpha\gamma} e(g,g)^{\beta\gamma\varepsilon}} =$$

$$\frac{m \cdot \left(e(g,g)^{\sum_{i \in \{1,\ldots,x\}} \varepsilon\beta\lambda_i w_i}\right)}{e(g,g)^{\beta\gamma\varepsilon}} = m$$

5.2 Security analysis

(1) Anonymity: Vehicles use pseudo-identities instead of their real identities during the communication process, and the real identities of vehicles are stored in the non-attackable TPD, which effectively protects the privacy of their identities. Additionally, for a malicious vehicle, TA can obtain its real

identity according to its pseudo-identity so as to investigate the responsibility of this vehicle. The calculation formula is:

$$\begin{aligned} \mathrm{PID}_i^2 \oplus h\left(k_s \cdot \mathrm{PID}_i^1\right) &= \mathrm{RID} \oplus h(r_i \cdot K_s) \oplus h(k_s \cdot r_i \cdot g) \\ &= \mathrm{RID} \end{aligned}$$

(2) Message authentication: In our scheme, signing message ensures that the message will not be tampered with during transmission, so the integrity of the message and the legitimacy of the message owner are guaranteed.
(3) Data confidentiality: We use ECC to transmit the list of attributes L_B. The session key S for decrypting L_B can be calculated only when the private key k_A is known. However, according to ECDLP, we can know that it is difficult for other vehicles to get the private key. Therefore, the confidentiality of the data can be ensured.
(4) Unlinkability: Unlinkability is an effective complement to anonymity, which makes it impossible for a receiver to link one user who is interacting with it currently with another who was previously authenticated by it. Every time the sender sends a message, it needs to select a random number which will be used in the signature function. And some system parameters are safely stored in the TPD. Therefore, the malicious attacker cannot judge whether he has authenticated the same vehicle twice according to pseudo-identity.

5.3 Efficiency analysis

Our experiment was run on an Intel Core i3 2.4-GHz processor with MIRACL library [8] and Crypto++ library [9]. We compared our scheme with other two scheme [5, 45]. Some operations about execution time are defined as follows.

(1). T_{ab}: The execution time of a multiplication operation $ab \bmod n$, where $a, b \in Z_q^*$.
(2). T_{xP}: The execution time of a scale multiplication operation $x \cdot P$, where $x \in Z_q^*$ and $P \in E$.
(3). T_{g^x}: The execution time of a modular exponentiation $g^x \bmod n$, where $x \in Z_q^*$.
(4). T_{pair}: The execution time of a bilinear pairing operation $e(aP, bP)$, where $a, b \in Z_q^*$ and $P \in E$.
(5). T_{hash}: The execution time of SHA256 hash function operation.
(6). T_{AES}: The execution time of the encryption or decryption operation of AES-CCM algorithm.
(7). $T_{\mathrm{RSA-ED}}$: The execution time of the encryption or decryption operation of RSA1024 algorithm.
(8). $T_{\mathrm{RSA-SV}}$: The execution time of the signature or verification operation of RSA1024 algorithm.
(9). T_{ECIES}: The execution time of the encryption operation of elliptic curve integrate encrypt scheme.
(10). T_{ECDSA}: The execution time of the signature or verification operation of elliptic curve digital signature algorithm.

As we all know, it was difficult to measure accurately due to the short single-step execution time in the experiment. So, we choose more steps in the program and choose a longer input on the data to improve the accuracy of the measurement results. For the four operations of hashing, signing, encryption, and decryption, we set the number of for loops to 1000 and select the random bit string with the maximum length as the input. Then, the average value, dividing the time spent by 1000, is taken as the execution time of the operation. For the AES encryption/decryption algorithm, we use the counter with CBC-MAC mode. For the RSA encryption/decryption and sign/verification algorithm, we use the 1024-bit key and RSA encryption with PKCS v1.5 padding. For the ECIES and ECDSA algorithm, we use the secp256r1 as the initial parameter of the elliptic curve and SHA256 as the hash function.

All the parameters in the above operations including a, b, x, P are selected randomly from their domains of definition. Finally, we got the time cost of above operations from the experiment and listed them in Table 1.

Throughout the interaction process in our scheme, the whole time cost includes the time to encrypt, sign, verify, and decrypt. The time needed to perform the calculation operation $K_A = k_A G$ in step (1) of Section 4.4 is $T_{xP} = 1.258ms$. Then, the time to sign a message by the function $\mathrm{sig}(m) = k_s h(\mathrm{PID}_i) + r_i h(m \| T)$ is $T_{\mathrm{sign}} = 2T_{ab} = 0.0692ms$. Similarly, the time to verify the message that the vehicle receives by calculating the equation $\mathrm{sig}(m) \cdot G = K_s \cdot h(\mathrm{PID}_i) + \mathrm{PID}_i^1 \cdot h(m \| T)$ is $T_{\mathrm{ver}} = 3T_{xP} = 3.774ms$. In step (3) and step (4), the respective execution time of $R = rG$, $S = rK_A$, and $S = k_A R$ is also equal to $T_{xP} = 1.258ms$, and

Table 1 Execution time cost of different cryptographic operations

Operations	Times (ms)
T_{ab}	0.0346
T_{xP}	1.258
T_g^x	3.3421
T_{pair}	23.625
T_{hash}	0.005
T_{AES}	0.022
$T_{\mathrm{RSA-ED}}$	0.13/1.51
$T_{\mathrm{RSA-SV}}$	1.49/0.13
T_{ECIES}	4.35
T_{ECDSA}	3.01/8.89

the encryption process $c = \mathrm{ENC}_S(L_B)$ and decryption process $L_B = \mathrm{DEC}_S(c)$ take approximately $2T_{\mathrm{AES}} = 0.044ms$ in total. It should be noted that since the three comparison schemes all use the ABE algorithm identically, we have not taken into account the time overhead of this part. In a similar way, the total time overhead for the encryption and decryption operations of [45] is $2T_{\mathrm{pair}}$ and that of [5] is $2T_{\mathrm{pair}}5T_{ab} + 2T_{g^x} + 2T_{xP} + T_{\mathrm{RSA\text{-}ED}} + T_{\mathrm{RSA\text{-}SV}}$. As for the signature and verification operations, [45] needs to calculate the time overhead of signature generation, the certificate verification, and the message signature verification. And for [5], we use the same calculation method as ours to maintain consistency since its author does not specify the specific signature and verification method. The result is showed in Fig. 3.

5.4 Result and discussion

From Fig. 3, we can see that our scheme, requiring less execution time when transmitting the same number of messages, has better performance than the other two schemes.

The reason for such a result is that the elliptic curve encryption algorithm we use is more efficient than the other two papers' algorithms. As shown in Table 1, the time of the bilinear pair encryption algorithm used in [45] T_{pair} is much larger than T_{xP} and T_{AES} of our scheme. The execution time of a modular exponentiation T_{g^x} and that of the encryption or decryption operation of RSA used in [5] are also greater than our T_{xP} and T_{AES}. Therefore, our method has the best performance.

However, the main time overhead for our scheme is spent on the message signing and authentication operations, so its limitations will be clearly reflected when the number of vehicles participated in data sharing is greatly large.

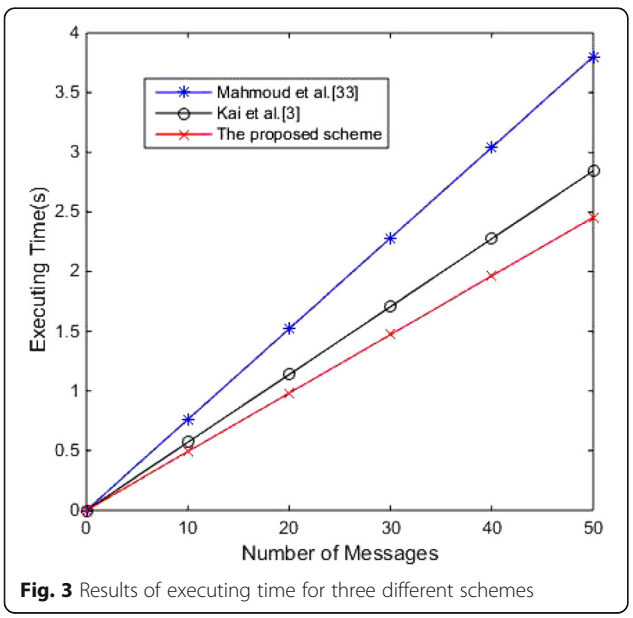

Fig. 3 Results of executing time for three different schemes

6 Conclusions

We propose a secure scheme based on edge computing to achieve data sharing among different domains. Next, we use ECC, CP-ABE, and the message authentication mechanism during the phase of data encryption and transmission. Finally, an analysis of our scheme demonstrates its security and efficiency.

In our scheme, the method used for selecting ECVs takes into consideration the number of available computing resources and their distances to the RSUs. In the future, we plan to include the social centrality of vehicles and select vehicles that can contact and interact with more vehicle nodes as edge computing nodes.

Abbreviations

AES: Advanced Encryption Standard; AES-CCM: Advanced Encryption Standard-Counter with CBC MAC; CA: Certification authority; CBC-MAC: Cipher Block Chaining Message Authentication Code; CLSS: Certificateless short signature; CPABE: Ciphertext Policy Attribute Based Encryption; CSP: Cloud service provider; CT: Cipher Text; ECC: Elliptic curve cryptography; ECDLP: Elliptic curve discrete logarithm problem; ECDSA: Elliptic curve digital signature algorithm; ECIES: Elliptic curve integrated encryption scheme; ECV: Edge computing vehicle; EG: Evolutionary games; IBS: Identity-based signature; IoT: Internet of Things; MAC: Media access control; MBS: Macro base station; MEC: Mobile edge computing; MK: Master key; OBU: On-board unit; OVU: Ordinary vehicle user; PID: Pseudo-identity; PK: Public key; PKCS: Public key cryptography standards; PKI: Public key infrastructure; PW-CPPA-GKA: Password-based conditional privacy protection authentication and group key generation; RID: Real identity; RSU: Roadside unit; SHA256: Secure Hash Algorithm 256; SPK: System public key; TA: Trusted authority; T_{ab}: The execution time of a multiplication operation; T_{AES}: The execution time of the encryption or decryption operation of AES-CCM algorithm; T_e: The execution time of a bilinear pairing operation; T_{ECDSA}: The execution time of the signature or verification operation of elliptic curve digital signature algorithm; T_{ECIES}: The execution time of the encryption operation of elliptic curve integrate encrypt scheme; T_g: The execution time of a modular exponentiation; T_h: The execution time of a hash function operation; TPD: Tamper-proof device; $T_{\mathrm{RSA\text{-}ED}}$: The execution time of the encryption or decryption operation of RSA1024 algorithm; $T_{\mathrm{RSA\text{-}SV}}$: The execution time of the signature or verification operation of RSA1024 algorithm; T_{xP}: The execution time of a scale multiplication operation; V2I: Vehicle-to-infrastructure; V2V: Vehicle-to-vehicle; VANETs: Vehicular ad hoc networks

Acknowledgements

The authors would like to thank Jing Zhang for her comments and suggestions.

Authors' contributions

JP carried out the study and drafted the manuscript. LW conceived the idea and participated in the design of the algorithm. JP and LW performed the experiment and analyzed the result. JC, YX, and HZ participated in the technical discussion and helped to perform the data analysis. All authors read and approved the final manuscript.

Funding

The work was supported by the National Natural Science Foundation of China (No. 61872001, No. 61572001, No. 61702005), the Open Fund of Key Laboratory of Embedded System and Service Computing (Tongji University), Ministry of Education (No. ESSCKF2018-03), the Open Fund for Discipline Construction, Institute of Physical Science and Information Technology, Anhui University, and the Excellent Talent Project of Anhui University.

Availability of data and materials

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Competing interests
The authors declare that they have no competing interests.

Received: 16 March 2019 Accepted: 10 June 2019
Published online: 24 June 2019

References

1. C.K. Toh, Ad hoc mobile wireless networks: protocols and systems. Pearson Education (2001)
2. J.J. Cheng, J.L. Cheng, M.C. Zhou, et al., Routing in internet of vehicles: A review[J]. IEEE Transactions on Intelligent Transportation Systems **16**(5), 2339–2352 (2015)
3. J. Cui, L. Wei, J. Zhang, et al., An efficient message-authentication scheme based on edge computing for vehicular ad hoc networks. IEEE Transactions on Intelligent Transportation Systems, 1–12 (2018)
4. J. Cui, H. Zhong, W. Luo, et al., Area-based mobile multicast group key management scheme for secure mobile cooperative sensing[J]. Science China(Information Sciences), 286–292 (2017)
5. K. Fan, Q. Pan, J. Wang, et al., *Cross-domain based data sharing scheme in cooperative edge computing* (2018 IEEE International Conference on Edge Computing, 2018), pp. 87–92
6. G. Luo, Q. Yuan, H. Zhou, et al., Cooperative vehicular content distribution in edge computing assisted 5G-VANET. China Communications **15**(7), 1–17 (2018)
7. X. Liu, R. Zhu, B. Jalaian, et al., Dynamic spectrum access algorithm based on game theory in cognitive radio networks. Mobile Networks and Applications **20**(6), 817–827 (2015)
8. Scott M, Multiprecision integer and rational arithmetic C/C++ library (MIRACL), (2003). https://www3.cs.stonybrook.edu/~algorith/implement/shamus/implement.shtml.
9. Dai W, Crypto++ library 5.1-a free C++ class library of cryptographic schemes, (2004). https://www.cryptopp.com/.
10. K. Zeng, Pseudonymous PKI for ubiquitous computing. European Public Key Infrastructure Workshop, 207–222 (2006)
11. X. Lin, X. Sun, P.H. Ho, et al., GSIS: a secure and privacypreserving protocol for vehicular communications. IEEE Transactions on Vehicular Technology **56**(6), 3442–3456 (2007)
12. C. Zhang, R. Lu, X. Lin, et al., in *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*. An efficient identity-based batch verification scheme for vehicular sensor networks (2008), pp. 246–250
13. S.J. Horng, S.F. Tzeng, P.H. Huang, et al., An efficient certificateless aggregate signature with conditional privacy-preserving for vehicular sensor networks. Information Sciences **317**, 48–66 (2015)
14. P. Vijayakumar, M. Azees, A. Kannan, et al., Dual authentication and key management techniques for secure data transmission in vehicular ad hoc networks. IEEE Transactions on Intelligent Transportation Systems **17**(4), 1015–1028 (2016)
15. M. Azees, P. Vijayakumar, L.J. Deboarh, EAAP: efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks. IEEE Transactions on Intelligent Transportation Systems **18**(9), 2467–2476 (2017)
16. J.L. Tsai, A new efficient certificateless short signature scheme using bilinear pairings. IEEE Systems Journal **11**(4), 2395–2402 (2017)
17. S.M. Pournaghi, B. Zahednejad, M. Bayat, et al., NECPPA: a novel and efficient conditional privacy-preserving authentication scheme for VANET. Computer Networks **134**, 78–92 (2018)
18. M.R. Asaar, M. Salmasizadeh, W. Susilo, et al., A secure and efficient authentication technique for vehicular ad-hoc networks. IEEE Transactions on Vehicular Technology **67**(6), 5409–5423 (2018)
19. Y. Liu, L. Wang, H.H. Chen, Message authentication using proxy vehicles in vehicular ad hoc networks. IEEE Transactions on Vehicular Technology **64**(8), 3697–3710 (2015)
20. S.K.H. Islam, M.S. Obaidat, P. Vijayakumar, et al., A robust and efficient password-based conditional privacy preserving authentication and group-key agreement protocol for VANETs. Future Generation Computer Systems **84**, 216–227 (2018)
21. J. Cui, J. Zhang, H. Zhong, et al., SPACF: A secure privacy-preserving authentication scheme for VANET with cuckoo filter. IEEE Transactions on Vehicular Technology **66**(11), 10283–10295 (2017)
22. J. Cui, J. Wen, S. Han, et al., Efficient privacy-preserving scheme for real-time location data in vehicular ad-hoc network. IEEE Internet of Things Journal **5**(5), 3491–3498 (2018)
23. H. Zhong, B. Huang, J. Cui, et al., *Efficient conditional privacy-preserving authentication scheme using revocation messages for VANET. 2018 27th International Conference on Computer Communication and Networks* (2018), pp. 1–8
24. T. Jing, Y. Pei, B. Zhang, et al., An efficient anonymous batch authentication scheme based on priority and cooperation for VANETs. EURASIP Journal on Wireless Communications and Networking **277** (2018)
25. H. Sago, M. Shinohara, T. Hara, et al., in *21st International Conference on Advanced Information Networking and Applications Workshops*. A data dissemination method for information sharing based on inter-vehicle communication, vol 2 (2007), pp. 743–748
26. Y. Zhang, J. Zhao, G. Cao, Roadcast: a popularity aware content sharing scheme in VANETs. ACM SIGMOBILE Mobile Computing and Communications Review **13**(4), 1–14 (2010)
27. Y. Zhu, Y. Zhang, X. Li, et al., Improved collusion-resisting secure nearest neighbor query over encrypted data in cloud. Concurrency and Computation: Practice and Experience, e4681 (2018)
28. X. Li, Y. Zhu, J. Wang, et al., On the soundness and security of privacy-preserving SVM for outsourcing data classification. IEEE Transactions on Dependable and Secure Computing, 1–1 (2017)
29. J. Xu, D. Zhang, L. Liu, et al., Dynamic authentication for cross-realm SOA-based business processes. IEEE Transactions on services computing **5**(1), 20–32 (2012)
30. J. Wang, R. Zhu, S. Liu, A differentially private unscented Kalman filter for streaming data in IoT. IEEE Access **6**, 6487–6495 (2018)
31. Y. Hao, J. Tang, Y. Cheng, Secure cooperative data downloading in vehicular ad hoc networks. IEEE Journal on Selected Areas in Communications **31**(9), 523–537 (2013)
32. D. Wu, H. Liu, Y. Bi, et al., Evolutionary game theoretic modeling and repetition of media distributed shared in P2P-based VANET. International Journal of Distributed Sensor Networks **10**(6), 718639 (2014)
33. Y. Lai, L. Zheng, T. Wang, et al., in *International Conference on Security, Privacy, and Anonymity in Computation, Communication, and Storage*. Cloud-assisted data storage and query processing at vehicular ad-hoc sensor networks (2017), pp. 692–702
34. J. Li, Y. Jia, L. Liu, et al., CyberLiveApp: A secure sharing and migration approach for live virtual desktop applications in a cloud environment. Future Generation Computer Systems **29**(1), 330–340 (2013)
35. D. Miao, L. Liu, R. Xu, et al., An efficient indexing model for the fog layer of industrial internet of things. IEEE Transactions on Industrial Informatics **14**(10), 4487–4496 (2018)
36. W. Shi, S. Dustdar, The promise of edge computing. Computer **49**(5), 78–81 (2016)
37. W. Shi, J. Cao, Q. Zhang, et al., Edge computing: vision and challenges. IEEE Internet of Things Journal **3**(5), 637–646 (2016)
38. Y. Mao, C. You, J. Zhang, et al., A survey on mobile edge computing: the communication perspective. IEEE Communications Surveys & Tutorials **19**(4), 2322–2358 (2017)
39. J. Ren, H. Guo, C. Xu, et al., Serving at the edge: a scalable IoT architecture based on transparent computing. IEEE Network **31**(5), 96–105 (2017)
40. R. Roman, J. Lopez, M. Mambo, Mobile edge computing, Fog et al.: a survey and analysis of security threats and challenges. Future Generation Computer Systems **78**, 680–698 (2018)
41. Q. Yuan, H. Zhou, J. Li, et al., Toward efficient content delivery for automated driving services: an edge computing solution. IEEE Network **32**(1), 80–86 (2018)
42. V.S. Miller, Use of elliptic curves in cryptography. Conference on the theory and application of cryptographic techniques, 417–426 (1985)
43. N. Koblitz, Elliptic curve cryptosystems. Mathematics of computation **48**(177), 203–209 (1987)
44. J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption. IEEE Symposium on Security and Privacy. IEEE Computer Society, 321–334 (2007)
45. M.H. Eiza, Q. Ni, Q. Shi, Secure and privacy-aware cloud-assisted video reporting service in 5G-enabled vehicular networks. IEEE Transactions on Vehicular Technology **65**(10), 7868–7881 (2016)

7 Publisher's Note