Secure Real-Time Traffic Data Aggregation Scheme Based on Privacy Computing in IoV

Jie Cui[a], Can Liao[a], Jing Zhang[a], Lu Wei[a], Hong Zhong[a], Irina Bolodurina[b]
[a] School of Computer Science and Technology, Anhui University, Hefei, China. [b] Faculty of Mathematics and Information Technologies, Orenburg State University, Orenburg, Russia
Cuijie@mail.ustc.edu.cn, {e21201106, zjing, weilu, zhongh}@ahu.edu.cn , prmat@mail.osu.ru

***Abstract*—As the intelligent transportation systems develop, vehicles are equipped with more sensing units, resulting in enormous sensing data being generated. Analytically computing these data has become crucial for enhancing the performance of intelligent transportation systems and optimizing energy efficiency. Currently, the challenge lies in performing privacy-preserving computations on traffic data while ensuring their confidentiality and improving the efficiency of data aggregation algorithms. Existing algorithms have various shortcomings such as inefficiency, a limited guarantee in the legitimacy of data sources, and a failure to apply access control to the aggregated data. To address these challenges, this study proposes a secure and efficient traffic data aggregation scheme in Internet of Vehicles (IoV). This scheme authenticates the identities of vehicles that send data, thereby enhancing the legitimacy and reliability of data sources. In contrast to prior studies that relied on centralized frameworks, we design a data aggregation protocol based on a distributed two-trapdoor public-key cryptosystem, which can compute statistical functions such as sum, average, variance, and maximum/minimum. Furthermore, the integration of an attribute-based encryption scheme is used to apply access control to the aggregated results. The proposed framework and protocol are evaluated; they outperform existing schemes in security properties and computational overhead.**

***Index Terms*—privacy computing, identity privacy, traffic data aggregation, IoV, homomorphic encryption**

I. Introduction

Nowadays, vehicles are becoming increasingly intelligent and are equipped with more sensing units capable of collecting information such as speed, position, temperature, images, videos, and exhaust emission data [1], [2]. These data are shared or uploaded for cloud-based analysis and calculation to enhance transportation efficiency, improve emergency responses, and reduce pollution and energy consumption. With the development of wireless technologies and network infrastructures. The Internet of Vehicles (IoV) enables cooperative processing of this information through infrastructures like road-side units (RSUs) and base stations (BSs) [3]. IoV uses data aggregation techniques to compile reliable data from multiple sources, significantly enhancing the design and functionality of intelligent transportation systems.

To ensure the privacy and availability of the original data and realize secure data aggregation, data owners usually employ homomorphic encryption algorithms (e.g., the Paillier cryptosystem [4] and the Brakerski-Gentry-Vaikuntanathan (BGV) encryption scheme [5]) and then send the encrypted data to the cloud server (CS). The CS performs secure computation and aggregation on the data using homomorphic encryption features. However, performing data aggregation involves several challenges, which are described below.

Although some schemes provide secure mechanisms for data aggregation [6], they do not authenticate the mobile devices that provide data. The presence of malicious vehicles poses a significant challenge in obtaining high-quality data. These devices not only degrade data quality, but also cause serious personal and material losses [7]. Furthermore, secure computation approaches typically rely on one or two cloud servers for data aggregation [8], [9], resulting in a potential single point of failure. The current schemes inherently lack support for applying multi-user access to the computational outcomes of encrypted data. Although attribute-based encryption (ABE) provides a powerful solution and has been widely applied in various application scenarios [10], [11], applying access control to data processing results remains a difficult problem [12]. In prior study [13], the problem was attempted to be resolved through a combination of homomorphic encryption and proxy re-encryption; however, this scheme only supported single access requests. If multiple users intend to access the same aggregated result, the designed scheme needs to be executed individually, which will result in high communication and computation expenses.

In this study, we introduce a novel scheme to address these challenges. To reduce the participation of malicious vehicles and obtain high-quality traffic data, we perform identity verification for the vehicles (mobile devices) with privacy protection before aggregating the data sent by the vehicles. Based on the IoV model, RSUs are dispersedly distributed in various regions. We design a framework that employs RSUs to cooperate with the CS in the aggregation process and complete the aggregation task together. By employing ABE, we ensure that multiple users can access the aggregation results with the aggregation task being executed only once and that access is only being provided to specific authorized users. The primary contributions of the proposed scheme are outlined below.

- We design a traffic data collection framework with an RSU as the data collection entity, which seamlessly supports the secure computation of encrypted data from vehicles with CS. The framework not only preserves the privacy of the vehicle through secure authentication, but also ensures the integrity and confidentiality of the vehicle data.

979-8-3503-5171-2/24/$31.00 ©2024 IEEE

TABLE I COMPARISON OF RELATED WORKS

schemes	Identity authentication	Data Confidentiality	Identity Privacy	Access Control	Sum aggregation	Average aggregation	Variance aggregation	Min/Max aggregation
[12]	×	×	×	✓	✓	×	×	×
[14]	×	×	×	×	✓	×	✓	✓
[15]	×	×	×	×	✓	✓	✓	×
[16]	×	×	×	×	✓	✓	✓	×
[17]	×	✓	✓	×	✓	✓	✓	✓
[18]	×	×	×	×	✓	✓	✓	×
[19]	×	×	✓	×	✓	×	×	×
proposed scheme	✓	✓	✓	✓	✓	✓	✓	✓

- We design secure data aggregation protocols that enable the computation of representative statistics, such as the sum, average, variance, and minimum/maximum of encrypted data, while maintaining a high level of efficiency.
- We integrate ABE algorithm into our scheme to apply fine-grained access control to the aggregated results. This allows aggregated data to be accessed in a more controlled and securer manner.

II. RELATED WORK

Several existing schemes have introduced innovative approaches for data aggregation in fields such as power grids [20], vehicle-to-grid networks [21], wireless sensor networks [22], and mobile crowd sensing [23], [24], [25]. To address privacy concerns, Lu et al. [26] proposed a privacy-preserving data aggregation scheme for vehicle sensing systems that ensures privacy, data accuracy, and scalability. Fan et al. [27] developed a mobile sensing scheme focusing on privacy and trustworthiness, using value vector analysis to detect malicious data. He et al. [28] introduced a secure consensus-based protocol that achieves precise sum aggregation while preserving sensitive data confidentiality. Tang et al. [29] proposed a health data aggregation scheme that emphasizes privacy, securely aggregating data from multiple sources and offering fair incentives. However, these schemes still have limitations in supporting certain types of aggregation.

To address the limitation of supporting aggregation types, Zhuo et al. [14] proposed a verifiable data aggregation architecture for mobile crowdsourcing that emphasizes privacy preservation, though it demands high computational costs for pairing operations. To mitigate this, a new scheme [15] was introduced, supporting multiple aggregation types while ensuring both identity and data privacy.

Ganjavi et al. [23] proposed an edge-assisted mobile crowdsensing scheme to ensure participant privacy and protect against adversaries, but it lacks differential privacy integration, reducing data security. Yang et al. [24] introduced a data aggregation scheme using differential privacy, though it compromises between privacy cost and accuracy. Wang et al. [25] addressed this by proposing a spatial ciphertext aggregation scheme with fog nodes, though it only supports additive aggregation. Wu et al. [17] further developed a spatial ciphertext architecture utilizing fog computing, where distributed fog nodes collaborate with the SC-server for privacy-aware data aggregation.

Xu et al. [30] developed a user-centric attribute-based access control system to safeguard user data managed by cloud service providers. Cao et al. [31] achieved fine-grained access control at the dimension level, while Ding et al. [12] introduced a privacy-preserving data processing scheme with flexible access control. Zhao et al. [18] proposed a verifiable multi-dimensional encrypted medical data aggregation scheme for cloud-based WBANs, though it supports limited aggregation types. Ma et al. [19] introduced an edge subgroup data aggregation scheme where nodes verify data credibility using supervisory information. While ensuring privacy and integrity, the scheme's revocation messages may increase network load and latency, affecting real-time efficiency.

Table I summarizes the functionality achieved by related data aggregation schemes. Overall, previous approaches have assumed that data providers are trustworthy and that authentication during aggregation is unnecessary. However, real-world scenarios often involve malicious vehicles, and unverified data providers might falsify data. There is a lack of schemes for controlling access to aggregated results, which is crucial for ensuring security, privacy, and cost-efficiency. To address these problems, our scheme presents a novel data aggregation scheme that supports various aggregation types and implements flexible access control. This scheme ensures data authenticity and privacy, filling the gap in existing research and providing a more secure and efficient solution for traffic data aggregation, particularly with untrusted data providers.

III. PRELIMINARIES AND SYSTEM MODEL

A. Cryptography Primitives

1) Shamir's (t, n) Threshold Secret Sharing Scheme

Assuming that n participants, denoted as P_1, P_2, ... , P_n, have shared a secret S, the manager selects $t-1$ elements randomly, which are represented as a_1, a_2, ..., a_{t-1}, and builds a polynomial $f(x) = a_0 + a_1\ x + a_2 x_2 + \ldots + a_{t-1} x_{t-1}$ to share secret S. The manager utilizes the polynomial $f(x)$ to compute: $y_i = f(x_i)(i = 1, 2, \ldots, n)$, the resulting value y_i is then securely transmitted to participant P_i through a secure communication channel. Given n participants, any subset of t participants (let us assume them as P_1, P_2, ..., P_t) can utilize their corresponding y_i values to derive a point: (x_1, y_1), (x_2, y_2), ... ,(x_t, y_t), at the same time, by applying the Lagrange interpolation formula, a polynomial $f(x)$ is constructed, and the secret S is determined as $S = f(0)$. The formula for calculating $f(x)$ is presented below:

$$f(x) = \sum_{i=1}^{t} y_i \prod_{1 \leq j \leq t, i \neq j} \frac{x - x_i}{x_j - x_i} \quad (1)$$

If fewer than t out of n users attempt to retrieve the secret S using their individual secret shares, they will not be able to recover it.

2) Ciphertext-Policy Attribute-Based Encryption (CP-ABE)

$Setup_{ABE} \rightarrow (PK, MSK)$: To start the algorithm, the system initially chooses a bilinear group G_0 with prime order p and a generator g_a. Two random α' and β' are also selected from the set Z_p. The output of the algorithm consists of a public key PK and a master secret key MSK.

$$\begin{aligned} &PK = (G_0, g_a, h = {g_a}^{\beta'}, f = {g_a}^{1/\beta'}, e(g_a, g_a)^{\alpha'}). \\ &MSK = (\beta', {g_a}^{\alpha'}). \end{aligned} \quad (2)$$

$Enc_{ABE}(M, T, PK) \rightarrow CK$: Given the input message M, access policy T, and public key PK, algorithm proceeds by selecting a polynomial q_x for every node in the access tree, except the root node x. The value of $q_x(0)$ is set according to $Eq.(2)$. The output of the algorithm is the ciphertext CK, which is obtained using $Eq.(3)$. Here, s is a randomly generated number, Y represents the set of n leaf nodes in T.

$$q_x(0) = q_{parent(x)}(index(x)) \quad (3)$$

$$\begin{aligned} &CK = (\tau, \acute{C} = Me(g_a, g_a)^{\alpha' s}, C = h^s, \\ &\forall y \in Y : Cy = g_a^{q_y(0)}, Cy' = H(att(y))^{qy(0)}). \end{aligned} \quad (4)$$

$KeyGen_{ABE}(MSK, S) \rightarrow SK$: Given the input of the main secret key MSK and an attribute set S, the key generation algorithm produces a corresponding secret key SK.

$$\begin{aligned} SK = &(D = {g_a}^{(\alpha'+r)/\beta'}, \forall j \in S : Dj = {g_a}^{r}. \\ &H(j)^{r_j}, Dj' = {g_a}^{r_j}). \end{aligned} \quad (5)$$

$Dec_{ABE}(PK, SK, CK) \rightarrow M$: The decryption algorithm is able to decrypt the ciphertext CK and recover the original message M only if the attributes linked with the private key SK satisfy the access policy of the ciphertext.

If two pieces of data are encrypted using the consistent access policy, CP-ABE exhibits multiplicatively homomorphic property.

$$\begin{aligned} &\mathrm{Enc}^{\mathrm{ABE}}(M_a * M_b, \tau, PK) \\ &= \mathrm{Enc}^{\mathrm{ABE}}(M_a, \tau, PK) * \mathrm{Enc}^{\mathrm{ABE}}(M_b, \tau, PK) \end{aligned} \quad (6)$$

3) Distributed Two Trapdoor Public-Key Cryptosystem (DT-PKC) [32]

DT-PKC is a cryptographic primitive that possesses additive homomorphic property and allows for secure computations in a multi-key environment (i.e., with multiple users).

$KeyGen$: Assuming a security parameter k and two large prime numbers p and q, where p and q are two large prime numbers with k bits, we select two strong primes $p' = (p - 1)/2$ and $q' = (q - 1)/2$. Then, we calculate $N = pq$ and $\lambda = lcm(p - 1, q - 1)/2$, and define a function $L(x) = (x - 1)/N$. We choose a generator g_1 of order $(p - 1)(q - 1)/2$ and select random values $\theta_i \in [1, N/4]$ for each entity i, then compute $h_i = {g_1}^{\theta_i} \bmod N^2$. The public key for entity i is $pk_i = (N, g_1, h_i)$ and the corresponding weak private key is $sk_i = \theta_i$. The system's strong private key is $SK = \lambda$.

$SkeyS(\lambda)$: The algorithm divides the strong private key $SK = \lambda$ into two parts, $SK_j = \lambda_j (j = 1, 2)$, such that $\lambda_1 + \lambda_2 \equiv 0 \, mod \, \lambda$ and $\lambda_1 + \lambda_2 \equiv 1 \, mod \, N^2$.

$Enc(m, pk_i)$: This algorithm generates a ciphertext $[m] \, pk_i$ for a message m in Z_N. Firstly, a random number r is chosen from the range $[1, N/4]$. The ciphertext $[m] \, pk_i$ is then output as $(C_{i,1}$,$C_{i,2})$, where $C_{i,1} = {g_1}^{r\theta_i} (1 + mN) \ mod \, N^2$ and $C_{i,2} = {g_1}^{r} \ mod \, N^2$.

$WDec([m]_{pk_i}, sk_i)$: This algorithm takes as input a ciphertext $[m]pk_i$ and the corresponding weak private key $sk_i = \theta_i$, then uses it to obtain the plaintext m as $m = L((C_{i,1}/C^{\theta_i}{}_{i,2}) \, mod \, N^2)$.

$PSD1([m]_{pk_i}, \lambda_1)$: Given a ciphertext $[m] \, pk_i = (C_{i,1}, C_{i,2})$ and the partial strong private key λ_1, the first-step partial decryption process in this algorithm involves computing $CT_i^{(1)} = (C_{i,1})^{\lambda_1} = {g_1}^{r\theta_i\lambda_1}(1 + mN\lambda_1) \, mod \, N^2$.

$PSD2([m] \, pk_i, CT_i^{(1)}, \lambda_2)$: Given the partial decrypted ciphertext $CT_i^{(1)}$ and another partial strong private key λ_2, the second-step partial decryption process in this algorithm involves computing $CT_i^{(2)} = (C_{i,1})^{\lambda_2} = {g_1}^{r\theta_i\lambda_2}(1 + mN\lambda_2) \, mod \, N^2)$ and then then computes $m = L(CT_i^{(1)} * CT_i^{(2)})$.

Additive homomorphic property:

$$\begin{aligned} &[x_1]pk \cdot [x_2]pk = ((1 + (x_1 + x_2)N) \cdot h^{r_1 + r_2} mod N^2, \\ &{g_1}^{r_1 + r_2} mod N^2) = [x_1 + x_2]pk. \\ &([x]pk)^{N-1} = ((1 + (N - 1)xN)h^{(N-1)r_1} mod N^2, \\ &{g_1}^{(N-1)r_1} mod N^2) = [-x]pk. \end{aligned} \quad (7)$$

B. System Model

The proposed data aggregation scheme is based on a system model shown in Fig.1.

1) TA: Completely trustworthy , responsible for guiding the entire system, registering vehicles and RSUs, assigning keys to them, and running some necessary algorithms.
2) Vehicle: Entities that flexibly transmit data in accordance with their own will. To protect their data privacy, the data generated by vehicles is already encrypted before being submitted to the RSU.
3) RSU: Semi-trusted entity, located at the network edge that acts as a mediator between the CS and vehicles. Its responsibility is to retain and manage specific interaction parameters that have been produced by entity TA,

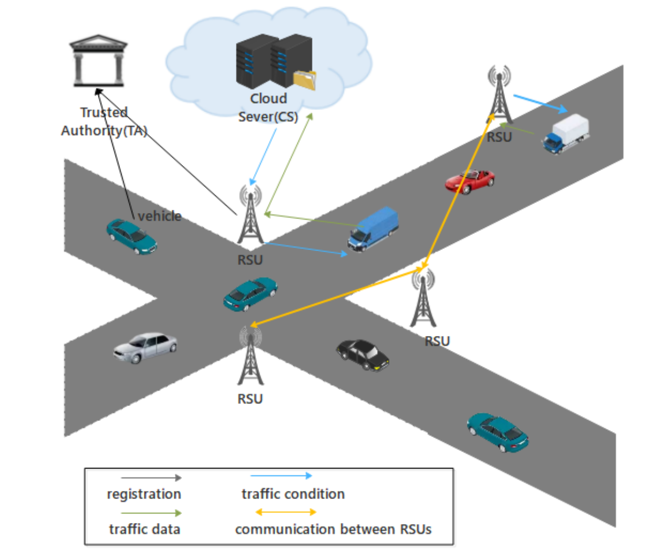


Fig. 1. system model

which are utilized for communicating among vehicles and RSUs, as well as for accumulating, storing, transmitting, and computing data obtained from vehicles.

4) CS: Cloud server, semi-trusted entity, works in conjunction with RSU (Fog Node), performs various operations on the collected data flexibly, and sends the results to the data requester.
5) DR: Data requester, an entity that requests aggregate results from the system.

C. Security and Privacy Adversaries

1) Identity Spoofing and Impersonation: Malicious actors may attempt to impersonate legitimate entities, such as vehicles or RSUs, to gain unauthorized access to the system. These attackers may include hackers seeking to steal identity information or masquerade as legitimate vehicles.
2) Key Leakage and Forgery: Adversaries might exploit leaked cryptographic keys to forge identities or manipulate communication. Potential adversaries include insider threats, such as external hackers with access to compromised keys.
3) Identity Linking and Tracking: Attackers may seek to link a vehicle's identity with its behavior to track and monitor it.
4) Data Tampering and Integrity Attacks: Adversaries may attempt to alter transmitted data to disrupt system operations or mislead decision-making processes. These attackers could include hackers, or entities with malicious intent.
5) Unauthorized Data Access and Misuse: Attackers could seek to access private data beyond their authorization, leading to excessive surveillance or improper use. Adversaries in this context may include data thieves, external requesters with malicious intent, and RSU (or CS) abusing their privileges.

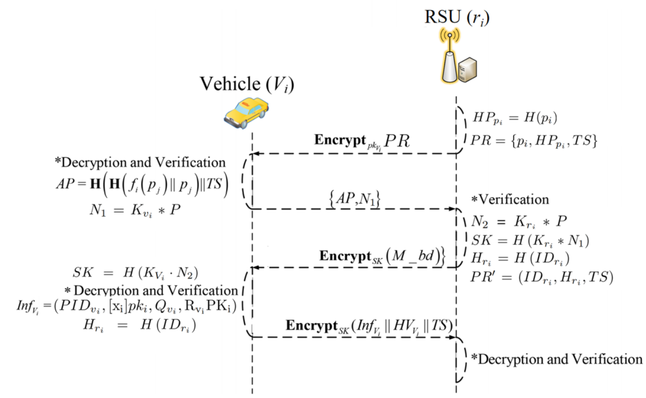


Fig. 2. Data interaction between vehicles and RSUs.

IV. PROPOSED SECURE TRAFFIC DATA AGGREGATION (STDA) SCHEME

A. Notations

For convenience of presentation, Table II provides an overview of the notations utilized in this paper.

B. System initialization

In this section, TA produces a sequence of public parameters that are used in the system, and registers RSUs and vehicles for subsequent authentication.

System initialization and entity registration are run by TA. This algorithm takes the security parameter k as input and outputs the system parameter $sp = (q_1, q_2, N, g_1)$ for DT-PKC, where q_1 and q_2 are two large prime numbers with a length of k bits, $N = q_1 q_2$, and g_1 is a generator of order $(q_1 - 1)(q_2 - 1)/2$. TA generates a group G_1 of order q whose generator is P_1, and selects a elliptic curve $Eq(a', b')$ over

TABLE II NOTATION DESCRIPTION

g_1	The system generator in DT-PKC
g_a	The system generator in ABE
H	A cryptographic hash function
λ	The strong private key in DT-PKC
λ_1, λ_2	The partial strong private keys
sk_i	The weak private key of vehicles in DT-PKC
pk_i	The public key of vehicles in DT-PKC
v_i, r_i	The serial number of vehicle and RSU in the system
ID_{v_i}	The real identity of v_i
PID_{v_i}	The pseudo identity of v_i
ID_{r_i}	The real identity of r_i
pk_E	The public key of DR
$[m_{v_i}]pk_i$	Ciphertext of message m_{v_i} encrypted by pk_i
α, β	Random numbers chosen by TA
CK_1,CK_2	The ciphertext of α, β encrypted by ABE
SK	The session key of vehicle and RSU
SK_i'	The private key in ABE

the finite field Z_q, where q is a large prime, and $a', b' \in Z_q$ are chosen in such a way that the condition $4a'^3 + 27b'^2 =$

$0(mod\, q)$ is met, P is a generator in $Eq(a', b')$ of an additive group G whose order is λ'.

TA generates $SK = \lambda = lcm(q_1 - 1, q_2 - 1)/2$ as the strong private key, and TA randomly selects $sk_i = \theta_i \in [1, N/4]$ as the weak private key of vehicles, the corresponding public key is $pk_i = (N, g_1, h_i = {g_1}^{\theta_i} modN^2)$. TA selects a random $d \in Z_N$, and sets the threshold R of all types of vehicles' data (d and R should be updated periodicity). Subsequently, algorithm $SkeyS(\lambda)$ is employed to split the strong private key $SK = \lambda$ into λ_1 and λ_2.

TA chooses two random numbers α, β where $\alpha \in [1, N/4], \beta \in [1, N/4], \alpha\beta \in [1, N/4]$ and $\theta_i + \alpha\beta \in [1, N/4]$, and runs the attribute encryption algorithm to get $CK_1 = E\mathrm{nc}(\alpha, \gamma, PK')$, where γ denotes access policy. Then, TA sets the encryption public key $PK_E = (N, g_1, h = {g_1}^{\theta_i + \alpha\beta})$. Cloud server generates the attribute key SK_i' for the data requester according to the attribute of DR through running the $KeyGen_{ABE}(MSK', S)$ algorithm. The θ_i, SK_i' is the group key of the DR. θ_i remains unchanged for a period of time, and will be updated with the update of access policy. Finally, TA sends $\{PK_E, \beta, R, \lambda_2\}$ to CS.

C. Entity registration

Vehicles registration: TA receives a vehicle registration request, it generates an ID_{v_i} and the corresponding pseudonym PID_{v_i}. $PID_{v_i} = \{ PID_{v_i,1}, PID_{v_i,2}\}$, $PID_{v_i,1} = k_i P_1$, $k_i \in Z^*{}_q$, $PID_{v_i,2} = ID_{v_i} \oplus H(\alpha_1 PID_{v_i,1}, PID_{v_i,1}, ET_i)$, random $\alpha_1 \in Z^*{}_q$, ET_i defines the valid period of this PID_{v_i}. TA selects private key sk_{v_i} for vehicles and generates the corresponding public key pk_{v_i}, meanwhile, TA runs the $KeyGen(sp)$ algorithm to genreate (sk_i, pk_i) for vehicles. TA generates a polynomial:

$$f_i(x) = a_0 + a_1 x + a_2 x^2 + ... + a_{t-2} x^{t-2} + a_{t-1} x_{t-1}$$

Where $a_0 = ID_{v_i}$, t is the threshold of Shamir Secret Sharing, TA chooses a random number R_{v_i}, TA sets $P_{v_i} = \{PID_{v_i}, f_i(x), R_{v_i}, d\}$, then uses pk_{v_i} to encrypt P_{v_i} and sends the encrypted result $(P_{v_i})pk_{v_i}$ to vehicles.

RSUs registration: TA selects private key sk_{r_i} for RSUs and generates the corresponding public key PK_{r_i}. TA generates the unique identity ID_{r_i} for RSU, and generates the safety parameter p_i, and generates the following parameters to communicate with the vehicle: $M_{r_i} = \{f_n(ID_{r_i}), HP_n, HPID_n\}$, where $HP_n = H(f_n(p_i) \parallel p_i), n \in [1, x], HPID_n = H(PID_{v_i})$, $n \in [1, x]$, F_n is a polynomial selected by TA for registered vehicles, x is the number of registered vehicles, and then a set of parameters P_{r_i} is generated by TA, P_{r_i} =$\{ID_{r_i}, p_i, \mathrm{d}, M_{r_i}\}$, then use pk_{r_i} to encrypts P_{r_i} and send Encrypted result $(P_{r_i})pk_{r_i}$ to RSUs. Finally, TA sends $\{CK_{1,} [d]\, pk_E, \lambda_1\}$to RSU.

D. Data collection

In this section, RSUs and the vehicles will conduct bidirectional authentication and data integrity verification to guarantee the integrity and credibility of the data source.

• When registered vehicles enter an RSU's management area, the interaction process for a vehicle V_i is as follows: The RSU decrypts the message from the TA, computes the hash value $HP_{p_i} = H(p_i)$, and then encrypts $PR = \{p_i, HP_{p_i}, TS\}$ using the vehicle's public key pk_{v_i}, where TS represents the current timestamp. The encrypted message is then sent to vehicle V_i.

• Using its private key sk_{v_i}, vehicle V_i decrypts the received ciphertext, checks the timestamp TS, and verifies HP_{p_i} by comparing it with $H(p_i)$. If PR is valid, V_i generates a random number K_{v_i}, calculates $N_1 = K_{v_i} * P$, and computes $AP = H(H(f_i(p_i) \parallel p_i \parallel TS)$, where TS is the received timestamp. Finally, V_i sends parameters N_1 and AP to RSU_{r_i}.

• Upon receiving N_1 and AP from V_i, RSU_{r_i} verifies the vehicle's identity by checking $H(HP_n \| TS) \stackrel{?}{=} AP$, where $n \in [1, t]$. If the verification is successful, RSU_{r_i} selects a random number K_{r_i}, computes $N_2 = K_{r_i} * P$, and derives the session key $SK = H(K_{r_i} * N_1)$. RSU_{r_i} then calculates $H_{r_i} = H(ID_{r_i})$ and generates $PR' = (ID_{r_i}, H_{r_i}, TS)$. Using the session key SK, RSU_{r_i} encrypts PR' and sends it to the vehicle along with N_2.

• Upon receiving the ciphertext and N_2, vehicle V_i computes the session key $SK = H(K_{v_i} \cdot N_2)$ and decrypts the ciphertext. It then retrieves ID_{r_i}, H_{r_i}, and TS, verifying the timestamp and checking if $H_{r_i} = H(ID_{r_i})$ is correct to validate the data block. If valid, V_i constructs its traffic message: $Inf_{v_i} = (PID_{v_i}, [x_i]pk_i, Q_{v_i}, R_{v_i}, pk_i)$, where $[x_i]pk_i = [m_{v_i} + d]_{pk_i}$ is the ciphertext from the DT-PKC encryption. $Q_{v_i} = f_i(ID_{r_i}) + H(R_{v_i})$, and $H_{v_i} = H(Inf_{v_i})$ for integrity. Finally, V_i encrypts $\{Inf_{v_i}, H_{v_i}, TS\}$ with SK and sends the ciphertext to RSU_{r_i}.

• Upon receiving the ciphertext, RSU_{r_i} decrypts it to retrieve the data and validates the data block by checking the timestamp and verifying if H_{v_i} matches $H(Inf_{v_i})$. If valid, RSU_{r_i} checks if $Q_{v_i} = f_i(ID_{r_i}) + H(R_{v_i})$ and $H(PID_{v_i}) = HPID_{v_i}$. This twofold verification resists forgery attacks. Once the verification is complete, RSU_{r_i} accepts the traffic data $[x_i]pk_i$. The ciphertext $[x_i]pk_i$, masked traffic data, is then used for secure computation by subsequent RSUs and the CS. The data interaction between vehicles and RSUs is depicted in Fig. 2.

• After receiving data from vehicles, the RSU performs preliminary processing and forwards it to the CS. If illegal or malicious vehicles are detected, the RSU can use Shamir Secret-Sharing to recover the vehicle's real identity. Once identified, the system rejects further data from these vehicles. The process for recovering the vehicle's real identity is as follows:

$$f_i(x) = \sum_{i=1}^{t} f_n(ID\, v_i). \prod_{m=1, m\neq i}^{t} \frac{x - f_n(ID\, v_m)}{f_n(ID\, v_i) - f_n(ID\, v_m)} \tag{8}$$

The system selects t RSUs to compute $f_i(x)$, determining the vehicle's real identity as $ID_{v_i} = f_i(0)$. Upon identifying a malicious vehicle, the number of registered vehicles decreases to $x - 1$, and the TA updates M_{r_i}. The parameters corresponding to the vehicle with identity ID_{v_i} in M_{r_i} are revoked, preventing the vehicle from completing subsequent authentication. Consequently, the RSU will reject data from the vehicle.

E. Data request

When the DRs want to request data from the CS, they will send a data request message to the CS, and CS will use the current pk_E of DR (it will change periodically with the update of the access policy) to encrypt aggregation result and send it to the DR.

F. Data aggregation

Algorithm 1 illustrates the sum aggregation. Vehicles send masked data $[m_{v_i} + d]pk_i$ to RSUs for aggregation, with mask d protecting the data from exposure during aggregation by RSU and CS. The RSU partially decrypts the data using λ_1 and forwards it to the CS. The CS fully decrypts and aggregates the data using λ_2, then encrypts the result with $Enc(S, pk_E)$. The RSU applies $Enc_{ABE}(\beta, \gamma, PK')$ to obtain CK_2. The CS sends $[S]pk_E$ and CK_2 to the RSU, which then removes the mask to derive the final aggregation result and compute CK based on the operational properties.

$$\begin{aligned} &Enc_{ABE}(\alpha, \gamma, PK') * Enc_{ABE}(\beta, \gamma, PK') \\ &= Enc_{ABE}(\alpha\beta, \gamma, PK') \end{aligned} \tag{9}$$

finally, $[\sum_{v_i \in n} m_i]pk_E$ and CK are sent to DR.

Algorithm 2 outlines the multiplication of two numbers. The process begins with the RSU partially decrypting the numbers using λ_1. The CS then fully decrypts the data using λ_2, calculates $(m_1 + d) * (m_2 + d) = x_1'' * x_2'' = M$, and encrypts M with pk_E. Homomorphic operation properties are then applied to compute:

$$\begin{aligned} &[M]pk_E * [\mathrm{m}_1 + m_2]pk_E^{N-d} * [d]pk_E^{N-d} \\ &= [(\mathrm{m}_1 + d) * (m_2 + d) - d * (m_1 + m_2) - d^2]pk_E \\ &= [m_1 * m_2]pk_E \end{aligned} \tag{10}$$

Algorithm 1: Secure SUM Aggregation

Input: $\lambda_1, PK_E, [d]pk_E, CK_1, pk_i\ \lambda_2, \beta, R$
Output: $[\sum_{v_i \in N} m_{v_i}]pk_E, CK$

```
// operation of RSU:
for all certified vehicles do
    x'_i ← PSD1(x_i, λ_1);
    Send(x_i, x'_i) to CS;
end
// operation of CS:
x''_i ← PSD2(x_i, x'_i, λ_2);
S ← Σ_{v_i∈N} x''_i, [S]pk_E ← Enc(S);
CK_2 = Enc_ABE(β, γ, PK');
Send[S]pk_E, CK_2 to RSU
// operation of RSU:
[Σ_{v_i∈n} m_{v_i}]pk_E ← [S]pk_E * ([d]pk_E)^(N−n)
CK = CK_1 * CK_2
```

Algorithm 2: Secure Multiplication

Input: $\lambda_1, PK_E, [d]pk_E, CK_1, pk_i\ \lambda_2, \beta, R$
Output: $[M]pk_E, CK$

```
// operation of RSU:
x'_1 ← PSD1(x_1, λ_1);
x'_2 ← PSD1(x_2, λ_1);
// operation of CS
x''_1 ← PSD2(x'_1, λ_2);
x''_2 ← PSD2(x'_2, λ_2);
CK_2 = Enc_ABE(β, γ, PK');
[M']pk_E ← [x''_1 * x''_2]pk_E;
Send[M]pk_E to RSU.
// operation of RSU :
[m_1]pk_E = [x_1]pk_E * [d]pk_E^(N−1);
[m_2]pk_E = [x_2]pk_E * [d]pk_E^(N−1)
[m_1 + m_2]pk_E = [m_1]pk_E * [m_2]pk_E;
[M]pk_E ← [M']pk_E * [m_1 + m_2]pk_E^(N−d) * [d]pk_E^(N−d);
CK = CK_1 * CK_2
```

Algorithm 3: Secure Average Aggregation

Input: m_{v_i}, n, pk_E
Output: $[\bar{d}]pkE$

```
// operation of RSU and CS :
call the Algorithm 1 to get:
[D]pk_E ← [Σ_{v_i∈n} m_{v_i}]pk_E;
// operation of RSU :
x ← 1/|n|;
x ← (m, e);
compute [m]pk_E, [e]pk_E ← ([−e]pk_E)^(N−1);
// operation of RSU and CS :
[m']pk_E ← M([m]pk_E, [D]pk_E);
[d̄]pk_E ← ([m']pk_E, [e]pk_E);
```

Algorithm 3 illustrates the division process, where the input consists of data from n vehicles and the output is their average. First, the system uses Algorithm 1 to compute the sum of the data. Since the DT-PKC algorithm only supports integer encryption, division results in non-integer values, requiring the use of decimal floating-point representation. The reciprocal of the total vehicle count is expressed as $m * 10^e$, where e is negative. The system encrypts the inverse of e, denoted as $[-e]pk_E$, and converts the division into a multiplication. The final average is expressed as $m' * 10^e$, allowing m to exceed 10, unlike traditional floating-point notation.

Algorithm 4 illustrates the variance aggregation. Based on Algorithm 3, the average is expressed as $m' * 10^e$, where m' is equal to the average divided by 10^{-e}. Since m_{v_i}, the average, and d are of similar magnitudes, d is adjusted to match m' by computing $[d * 10^{-e}]pk_E$. Then, $M = [m']\, pk_E * [d * 10^{-e}]\, pk_E$ is calculated, with RSU partially decrypting M using λ_1. Similarly, the vehicle data, also masked, is partially decrypted by the RSU. After being sent to the CS, both the data and masks are completely decrypted, and the variance is calculated by subtracting and counteracting the masks of M'' and x''.

Algorithm 4: Secure Variance Aggregation

Input: $\lambda_1, PK_E, [d]pk_E, CK_1, pk_i\ \lambda_2, \beta, R$
Output: $\overline{V} = [\sum_{v_i \in n} (m_{v_i} - \overline{d})^2 / n]pk_E$

// operation of RSU :
$M \leftarrow [m']pk_E * [d * 10^{-e}]pk_E$;
$M' \leftarrow PSD1(M, \lambda_1)$;
for *all certified vehicles* **do**
  $x_i{}' \leftarrow PSD1(x_i, \lambda_1)$;
  $Send\ (M, M', x_i, x_i{}') to\ CS$;
end
// operation of CS :
$M'' \leftarrow PSD2(M, M', \lambda_2)$;
$x_i{}'' \leftarrow PSD2(x_i, x_i{}', \lambda_2)$;
$X_i \leftarrow [M'' - x_i{}'' * 10^{-e}]^2 pk_E$;
$Send(X_i, M'', x_i{}'', CK_2)\ to\ RSU$;
// operation of RSU :
$Y \leftarrow \prod_{v_i \in n} X_i$
$input\ (Y, n)\ and\ Call\ algorithm3\ to\ get\ variance\overline{V}$;
$[\bar{V}]pk_E \leftarrow ([m'']pk_E, [3e]pk_E)$;

$$
\begin{aligned}
Y &= \prod_{v_i \in n} X_i \\
&= X_1 X_2 ... X_n \\
&= [M'' - x_1{}'' * 10^{-e}]^2 pk_E [M'' - x_2{}'' * 10^{-e}]^2 pk_E ... \\
&\quad [M'' - x_n{}'' *\ 10^{-e}]^2 pk_E \\
&= \{[M'' - x_1{}'' * 10^{-e}]^2 + [M'' - x_2{}'' * 10^{-e}]^2 + ... \\
&\quad [M'' - x_n{}'' * 10^{-e}]^2\} pk_E
\end{aligned}
\tag{11}
$$

The variance can be obtained by taking Y and the number of data n as the input of algorithm 3. It is worth noting that the data calculated by this operation is 10^{-2e} times the real variance.

$$
\begin{aligned}
\overline{V} &= [\sum_{v_i \in n} (m_{v_i} - \overline{d})^2 / n]pk_E \\
&= ([m'']pk_E, [3e]pk_E)
\end{aligned}
\tag{12}
$$

Algorithm 5 describes the computation of the minimum value from a set of vehicle data. The RSU partially decrypts the data, while the CS performs complete decryption. Assuming m_{v_i} is the minimum value, $[m_{v_i}]pk_E$ is used in conjunction with $[m_{v_1}]pk_E$ as inputs to the SLT [32] algorithm. The output of this algorithm is:

1) $u' = 0$ if $m_{v_1} >= m_{v_i}$, m_{v_i} is the current minimum value.

$$
\begin{aligned}
&Z_1 = [m_{v_1} - m_{v_i}]pk_E, Z_2 = 0 * Z_1 = 0, \\
&X = [m_{v_i} + 0]pk_E = [m_{v_i}]pk_E
\end{aligned}
\tag{13}
$$

2) $u' = 1$, if $m_{v_1} < m_{v_i}$, m_{v_1} is the current minimum value.

$$
\begin{aligned}
&Z_1 = [m_{v_1} - m_{v_i}]\, pk_E, Z_2 = 1 * Z_1, \\
&X = [m_{v_i} + m_{v_1} - m_{v_i}]\, pk_E = [m_{v_1}]\, pk_E
\end{aligned}
\tag{14}
$$

If we want to get the maximum value of a set of data, just make $u' = 1$ ($m_{v_1} >= m_{v_i}$), and $u' = 0$($m_{v_1} < m_{v_i}$).

G. Data decryption

The last aggregated result and CK are stored in CS. When the DR requests the aggregated result. CS will send aggregation result and CK to DR. The DR will first use SK' to decrypt CK to get $\alpha\beta$, and use θ_i jointly to decrypt aggregation result to get the data they request.

Algorithm 5: Secure MIN/MAX Aggregation

Input: $\lambda_1, PK_E, [d]pk_E, CK_1, pk_i\ \lambda_2, \beta, R$
Output: $[Min]pk_E, CK$

// operation of RSU :
for *all certified vehicles* **do**
  $x_i' \leftarrow PSD1(x_i, \lambda_1)$;
  $Send(x_i, x_i')\ to\ CS$;
end
// operation of CS:
$x_i'' \leftarrow PSD2(x_i, x_i', \lambda_2)$;
$compute[x_i'']pk_E$;
$CK_2 = Enc_{ABE}(\beta, \gamma, PK')$;
Send $CK_2, [x_i'']pk_E$ to RSU;
RSU operation:
$CK = CK_1 * CK_2$;
// operation of RSU and CS:
$X \leftarrow [x_1]pk_E$;
for *i=1 to n* **do**
  $[u']pk_E \leftarrow SLT(X, [m_{v_i}]pk_E)$;
  $Z_1 \leftarrow X * ([m_{v_i}]pk_E)^{N-1}$;
  $Z_2 \leftarrow M([u']pk_E, Z_1)$;
  $X \leftarrow [m_{v_i}]pk_E * Z_2$;
end
$Min \leftarrow [X]pk_E * ([d]pk_E)^{N-1}$

V. SECURITY ANALYSIS AND SECURITY PROOF

A. Security Analysis

1) Identity Authentication and Privacy

The vehicles and RSUs perform two-way authentication. The RSU stores the matrix M_{r_i} generated by the TA. Before sending data, the vehicle v_i computes $AP = H(H(f_i(p_i) \parallel p_i \parallel TS))$, where p_i is the parameter from the RSU and TS is a timestamp. The RSU retrieves M_{r_i} and checks $H(HP_n \parallel TS) \stackrel{?}{=} AP$ for authentication. In the second interaction, the vehicle sends $\{Inf_{v_i}, H_{v_i}, TS\}$, where Inf_{v_i} includes $(PID_{v_i}, [m_{v_i} + d]_{pk_i}, Q_{v_i}, R_{v_i}, PK_i)$. The RSU verifies if $Q_{v_i} = f_i(ID_{r_i}) + H(R_{v_i})$ and $H(PID_{v_i}) = HPID_{v_i}$ to authenticate the vehicle. In our scheme, the TA generates both the vehicle's identity and a pseudonym. The vehicle's identity ID_{v_i} is concealed using a secret-sharing scheme. Each RSU, being semi-trusted, can recover the vehicle's identity only by solving the polynomial. Any RSU with fewer than t shares cannot independently recover the identity.

2) Data Integrity and Confidentiality

When TA sends parameters to vehicles and RSUs, and during interactions between vehicles and RSUs, data is encrypted using public key encryption or session keys to ensure confidentiality. Additionally, our scheme uses $H()$ to compute a hash value for each transmitted data. Both the data and its hash value are sent, with the hash ensuring data integrity. This approach guarantees both data integrity and confidentiality.

3) Identity Traceability and Revocation

When RSU detects illegal data from the vehicle, t RSUs will be selected to recover the true identity of the vehicle. Each RSU calculates its secret share value to recover the polynomial:

$$f_i(x) = \sum_{i=1}^{t} f_n(ID_{v_i}) * \prod_{m=1, m\neq i}^{t} \frac{x - f_n(ID\,v_m)}{f_n(ID\,v_i) - f_n(ID\,v_m)} \tag{15}$$

The real identity of the vehicle $ID_{v_i} = f_i(0)$. After tracking the real identity of the malicious vehicle. The corresponding parameters of the vehicle whose identity is ID_{v_i} in M_{r_i} will also be revoked. In this way, this malicious vehicle cannot complete the subsequent authentication.

4) Data Privacy

In the data aggregation process, the aggregated data is encrypted using the DT-PKC. The RSU and CS perform aggregation on the encrypted data without exposing the real data plaintext from the vehicles, thereby ensuring the privacy of the data.

5) Access Control

We use Attribute-Based Encryption (ABE) to enforce strict access control, ensuring that only users with the correct attributes can decrypt and access the data. TA sets the encryption public key $PK_E = (N, g_1, h = g_1^{\theta_i+\alpha\beta})$, allowing decryption only for data requesters with θ_i and $\alpha\beta$. To prevent key leakage, parameters α are encrypted by TA, and β by CS. Data requesters use these properties to decrypt and obtain $\alpha\beta$.

6) Resist Various Attacks

Replay Attacks: During data collection, each data packet exchanged between the RSU and vehicles is timestamped. The recipient verifies the data by checking whether the timestamp is within the valid range. If the timestamp is expired, the data packet will be discarded, thereby preventing replay attacks.Moreover, the system periodically updates the key θ_i to ensure that even if data is replayed, it cannot be decrypted or tampered with.

Collusion Attacks: We use Shamir secret sharing, where only RSUs exceeding a threshold number can recover the true identity of the vehicle. This approach prevents malicious RSUs from collaborating to compromise the security of the system.

Forgery Attacks: We employ hash functions to ensure data integrity, data is checked for integrity during transmission and processing to ensure it has not been tampered with or forged.

B. Privacy Model-Based Formal Analysis

Theorem 1. The security of the DT-PKC scheme in Section III relies on the difficulty of the DDH problem over Z_{N^2}, ensuring semantic security (proof details are in [32]).

Definition 1. (Security in the semi-honest model.) Protocol π_i for party P_i is secure if its execution image $\Pi_i(\pi)$, based on input a_i and output b_i, can be simulated such that the simulated distribution is indistinguishable from $\Pi_i(\pi)$. A detailed security definition is provided in [33].

We use a security model that ensures the ideal functionality against semi-honest (non-colluding) adversaries. This model involves four entities: CS, RSU, Vehicle, and DR. We design four simulators $Sim = (Sim_{CS}, Sim_{RSU}, Sim_{Vehicle}, Sim_{DR})$ to counter adversaries $\mathcal{A}_{CS}, \mathcal{A}_{RSU}, \mathcal{A}_{Vehicle}, \mathcal{A}_{DR}$ targeting CS, RSU, Vehicle, and DR, respectively.

Theorem 2. The Sum Aggregation Algorithm can securely obtain the plaintext of addition via computations on ciphertexts in the context of semi-honest (non-colluding) adversaries $\mathcal{A} = (\mathcal{A}_{CS}; \mathcal{A}_{RSU}; \mathcal{A}_{Vehicle}; \mathcal{A}_{DR})$.

Proof. We construct four independent simulators $(Sim_{CS}, Sim_{RSU}, Sim_{Vehicle}, Sim_{DR})$ and demonstrate security with two inputs (i.e., $N = 2$).

The view of $\mathcal{A}_{Vehicle}$ is the encrypted traffic data. The views of $\mathcal{A}_{Vehicle}$ in the real and the hypothetical executions are impossible to differentiate.

Sim_{RSU} simulates $\mathcal{A}_{RSU}$ by masking the ciphertext with random numbers d, running $PSD1$ to get x_i', and then using Sim_{CS} to obtain $[S]pk_E$ and CK_2. It computes $[m_{v_1} + m_{v_2}]pk_E$ and CK, and outputs x_i', $[S]pk_E$, CK_2, $[m_{v_1} + m_{v_2}]pk_E$, and CK to $\mathcal{A}_{RSU}$. If $\mathcal{A}_{RSU}$ replies with $\perp$, Sim_{RSU} returns $\perp$.

The view of $\mathcal{A}_{RSU}$ comprises the encrypted data and the partial decryption key. Due to the honesty of the challenged vehicles and the semantic security of the DT-PKC scheme. $\mathcal{A}_{RSU}$ obtains identical outputs in both real and ideal executions. Thus, the views of $\mathcal{A}_{RSU}$ in the real implementation is indistinguishable from the ideal implementation.

Sim_{CS} simulates $\mathcal{A}_{CS}$ by running $PSD2$ to obtain x_i'' and adding x_i'' to get S. It then calls ABE encryption to obtain CK_2 and sends $[S]pk_E$ and CK_2 to $\mathcal{A}_{CS}$. If $\mathcal{A}_{CS}$ replies with $\perp$, Sim_{CS} returns $\perp$. In both real and ideal executions, Sim_{CS} produces the same two ciphertext outputs, with security ensured by the semantic security of the DT-PKC scheme and ABE.

The view of $\mathcal{A}_{CS}$ includes masked data and the ciphertext of the partial decryption key. Due to the honesty of the challenged vehicles and the semantic security of the DT-PKC scheme, $\mathcal{A}_{CS}$ obtains identical outputs in both real and ideal executions. Therefore, $\mathcal{A}_{CS}$'s view in the real implementation is indistinguishable from the ideal implementation.

Sim_{DR} simulates $\mathcal{A}_{DR}$ as follows: it randomly chooses $[m]pk_E$ and decrypts it to obtain m, and then sends it to $\mathcal{A}_{Vehicle}$. If $\mathcal{A}_{Vehicle}$ replies with $\perp$, Sim_{DR} returns $\perp$.

The view of $\mathcal{A}_{DR}$ is the decrypted result without any additional information. However, the semantic security of the DT-PKC guarantees the security of both real and ideal eimplementations, and the views of $\mathcal{A}DR$ in both implementations are indistinguishable.

Regardless of how many times the adversary queries the simulator $\mathcal{A}_{DR}$, it remains challenging to obtain the original traffic data due to two factors: 1) The randomly selected data are unrelated to the original real data; 2) Exhaustive attacks are difficult because of the randomness of the selected numbers.

The security proofs for other operations follow a similar approach to that of sum aggregation, considering semi-honest and non-colluding adversaries $(\mathcal{A}_{CS}; \mathcal{A}_{RSU}; \mathcal{A}_{Vehicle}; \mathcal{A}_{DR})$.

VI. PERFORMANCE EVALUATION

Simulation setup. We performed the simulations in Java based on the JPBC library, BigInteger Class, and Lombok library, to execute the DT-PKC cryptosystem, ABE algorithm, and related encryption algorithm in our scheme.

In the data collection phase, we evaluated the computational and communication overhead of the RSU and vehicles. During the data aggregation phase, the aggregation operations were performed by the RSU and CS. We analyzed the time complexity of various aggregation algorithms and compared them with related approaches, highlighting the superiority of our proposed scheme.

To demonstrate the efficiency of our method, we conduct a theoretical analysis of STDA's secure data aggregation costs and perform extensive experiments comparing STDA with Wu $et\ al.$'s schemes [17]. In the data aggregation stage, the ABE algorithm is used for access control, introducing additional computational overhead for RSUs and CS. Although ABE increases the computational burden, it provides robust security and fine-grained access control. Most computations are handled by cloud servers, with setup and encryption processes executed once and spread over multiple requests. Additionally, by changing the data mask from r_i to d, we reduce the overhead in subsequent data aggregation operations. We analyze and compare the computational costs of the four operations in the aggregation phase with Wu $et\ al.$'s scheme [17].

TABLE III OPERATION TIME OF DIFFERENT OPERATIONS(MS)

Operation	Time	Operation	Time	Operation	Time
$Enc1$	2.467	$Dec1$	3.619	$Enc2$	1.235
$Dec2$	1.416	$Exp1$	10.731	Enc	6.124
$PSD1$	10.325	$PSD2$	16.567	$Exp2$	12.358
$WDec$	2.453	$Pair$	8.142		

Table III shows the computational costs associated with different operations. We evaluated the cost of data interaction between vehicles and RSUs, as depicted in Fig. 3, which shows that the vehicle's overhead is a major part of the total interaction cost due to data generation. As data size increases, the overall cost rises; for example, with a 50KB data size, the process takes about 650ms, which is acceptable for IoV. In our scheme, data aggregation occurs mainly between the RSU and the cloud server, minimizing network impact between vehicles.

Fig. 4 displays the computational overhead for RSUs and vehicles during aggregation tasks with 50 vehicles per task. With tasks ranging from 10 to 50 and aggregated data increasing from 500 to 2500, our scheme shows significantly lower computational overhead for both RSUs and vehicles compared to Wu et al.'s scheme. Additionally, the vehicle's overhead remains constant and minimal. As vehicles perform few encryption operations, their computational requirements are low, making the system robust across different vehicle types and connectivity qualities.

Cost of Sum Aggregation: The RSU first partially decrypts the masked data using $PSD1$, with cost increasing with data volume. It then encrypts the intermediate result S with Enc and removes the mask using $Exp2$. The Mul operation

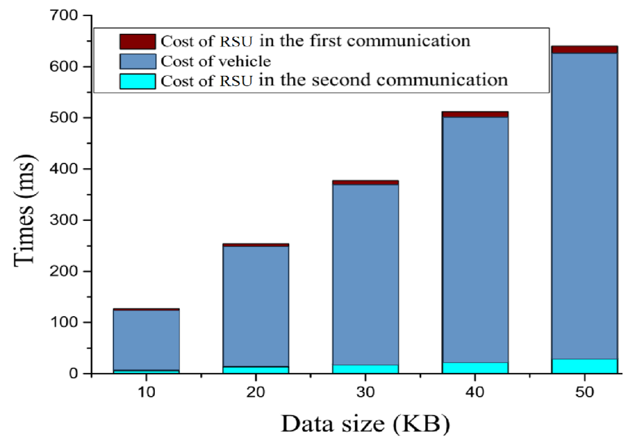


Fig. 3. The time cost of data collection process between a vehicle and RSU.

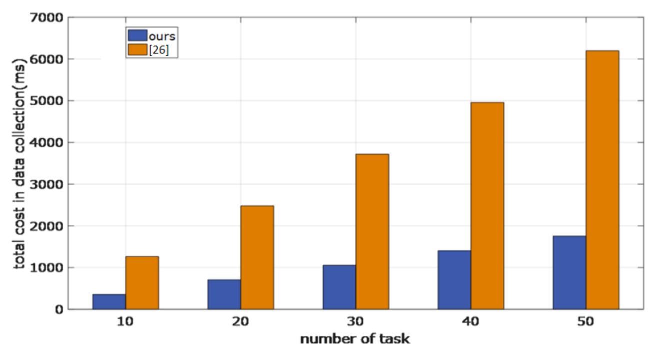


Fig. 4. The computational overhead of entities in the data collection phase

for ciphertext multiplication is negligible. The SC performs $PSD2$ to mask data, and then Enc and Enc_{ABE} to get the masked ciphertext. As shown in Fig. 2(a), CS incurs slightly higher overhead than SC due to additional access control and Enc_{ABE}.

Cost of Average Aggregation: Analyzing Algorithm 2 first, RSU performs $2\times PSD1$ and $3\times Exp2$. CS runs $2\times PSD2$ for masked data, Enc_{ABE} to encrypt β, and Enc for the result. Both RSU and CS call Algorithm 1, with RSU executing Enc and $Exp2$. The cost of calculating the average includes the sum computation cost, making it slightly higher than for the sum. Due to the access control, the CS cost in our scheme is somewhat higher compared to SC in Wu et al. [17].

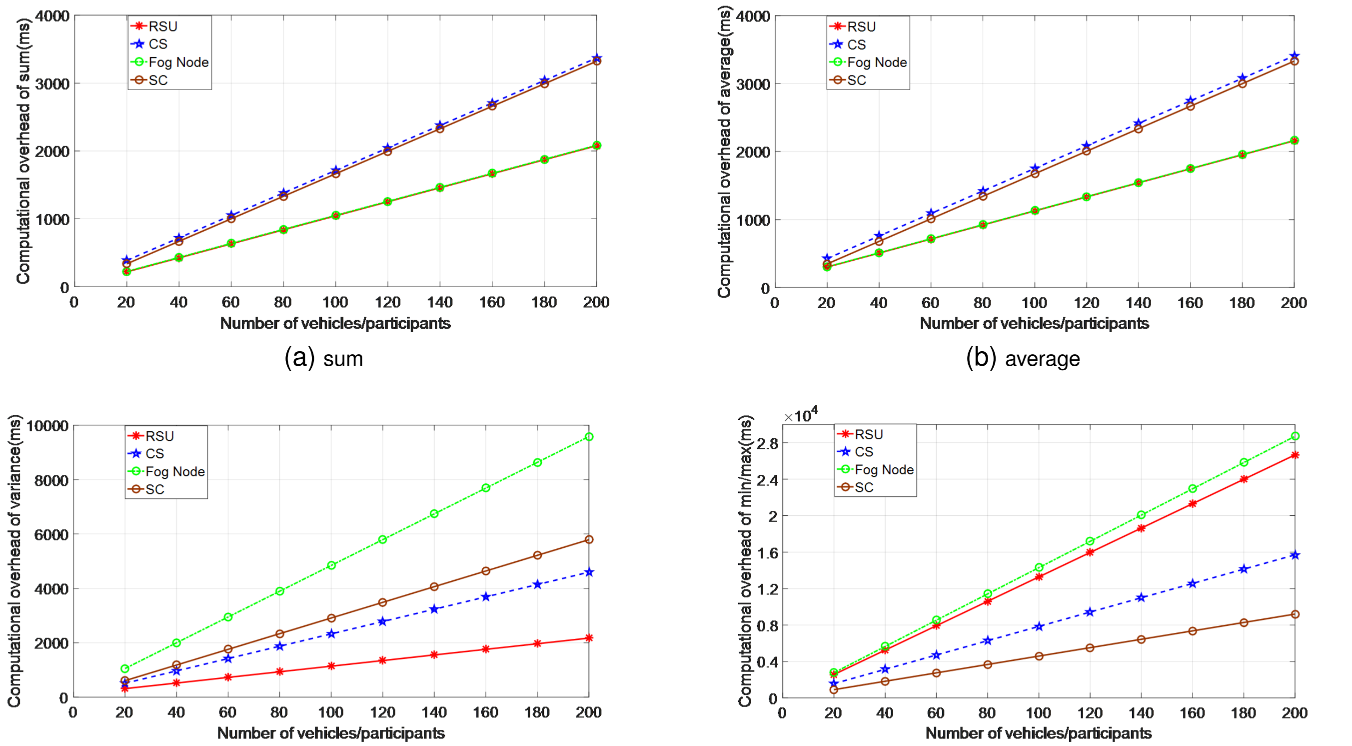
Cost of Variance Aggregation. Our scheme uses d to mask the original data, and TA encrypts d with pk_E and sends it to RSU together with d, which saves a lot of overhead when calculating variance. Fig. 2.(c) demonstrates that the expense of the RSU and CS in our scheme is less than that of the fog node and SC in Wu $et\ al.$ [17].

Cost of min/max Aggregation. RSU first needs to conduct n times $PSD1$ and once mul, CS needs to perform n times $PSD1$ and once Enc_{ABE}, then RSU and CS need to call SLT [32] and algorithm 2. In this phase, compared with Wu $et\ al.$ [17]. The cost of RSU in our STDA is lower than that of the fog node, and that of CS is higher than that of SC.

Scalability of our scheme. Impact of Real-World Testing

TABLE IV COMPARISON OF RSU, CS, FOG NODE AND SC CALCULATION OPERATIONS

	Sum	Average	Variance	Min/Max
RSU	$nPSD1+Exp2$	$(2+n)PSD1+5Exp2+2Enc$	$(n+3)PSD1+4Enc+4Exp2$	$nPSD1+Exp2+(n-1)(3PSD1+Enc+7Exp2)$
CS	$nPSD2+Enc+Enc_{ABE}$	$(n+2)PSD2+2Enc+Enc_{ABE}$	$(n+3)PSD2+(n+1)Enc+Enc_{ABE}$	$nPSD2+Enc+(n-1)(3PSD2+2Enc)+Enc_{ABE}$
Fog Node [17]	$nPSD1+Enc+Exp2$	$nPSD1+6Enc+5Exp2$	$(n+1)PSD1+7Enc+(3n+4)Exp2$	$nPSD1+nExp2+(n-1)(PSD1+4Enc+7Exp2)$
SC [17]	$nPSD2+Enc$	$nPSD2+2WDec+2Enc$	$(n+1)PSD2+(2n+1)Enc+2WDec$	$nPSD2+Enc+(n-1)(PSD2+2WDec+2Enc)$



(a) sum (b) average (c) variance (d) min/max

Fig. 5. Computational cost of different entities compared with [17].

Environment: Since most data aggregation is handled by cloud servers and RSUs, network fluctuations have minimal impact. We will examine communication overhead during data collection to illustrate minor real-world factors' effects.

Impact of network conditions: Data aggregation primarily occurs between RSUs and cloud servers. Vehicles only upload data to nearby RSUs, minimizing the effect of network conditions between vehicles on the aggregation process.

Impact of different vehicle types: Since vehicles perform few encryption operations and have low computational needs (Fig.4 shows), the impact of vehicle types on system performance is minimal.

VII. CONCLUSION

In this study, we proposes a real-time traffic data aggregation scheme based on distributed double trapdoor homomorphic encryption. In this scheme, vehicle authentication is conducted to ensure the reliability of data sources. This scheme employs the double trapdoor cryptographic system to design a secure and efficient data aggregation protocol. This protocol supports aggregation operations such as encrypted data summation, average, variance, and minimum/maximum. Access control is applied to aggregated results, enabling multiple users to access the results simultaneously, thus improving system efficiency. In the future, we will validate the practicality of our scheme through real-world application.

ACKNOWLEDGMENT

The work was supported by the National Natural Science Foundation of China under Grant U23A20308, Grant 62202008, Grant 62302008 and Grant 62325209.

References

[1] A. Arooj, M. S. Farooq, A. Akram, R. Iqbal, A. Sharma, and G. Dhiman, "Big data processing and analysis in internet of vehicles: architecture, taxonomy, and open research challenges," *Archives of Computational Methods in Engineering*, vol. 29, no. 2, pp. 793–829, 2022.

[2] B. Bhabani and J. Mahapatro, "Clurma: A cluster-based rsu-enabled message aggregation scheme for vehicular ad hoc networks," *Vehicular Communications*, vol. 39, p. 100564, 2023.

[3] M. Kaur, J. Malhotra, and P. D. Kaur, "A vanet-iot based accident detection and management system for the emergency rescue services in a smart city," in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, 2020, pp. 964–968.

[4] J. Zhang, Y. Zhao, J. Wu, and B. Chen, "Lvpda: A lightweight and verifiable privacy-preserving data aggregation scheme for edge-enabled iot," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4016–4027, 2020.

[5] G. Zhuo, Q. Jia, L. Guo, M. Li, and P. Li, "Privacy-preserving verifiable data aggregation and analysis for cloud-assisted mobile crowdsourcing," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, 2016, pp. 1–9.

[6] M. Girolami, E. Urselli, and S. Chessa, "Encrypted data aggregation in mobile crowdsensing based on differential privacy," in *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, 2022, pp. 22–25.

[7] K. Gu, X. Dong, and W. Jia, "Malicious node detection scheme based on correlation of data and network topology in fog computing-based vanets," *IEEE Transactions on Cloud Computing*, vol. 10, no. 2, pp. 1215–1232, 2020.

[8] X. Liu, B. Qin, R. H. Deng, and Y. Li, "An efficient privacy-preserving outsourced computation over public data," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 756–770, 2015.

[9] F. Rezaeibagha, Y. Mu, K. Huang, L. Chen, and L. Zhang, "Toward secure data computation and outsource for multi-user cloud-based iot," *IEEE Transactions on Cloud Computing*, vol. 11, no. 1, pp. 217–228, 2021.

[10] Z. Yan, X. Li, M. Wang, and A. V. Vasilakos, "Flexible data access control based on trust and reputation in cloud computing," *IEEE transactions on cloud Computing*, vol. 5, no. 3, pp. 485–498, 2015.

[11] Z. Yan, X. Li, and R. Kantola, "Controlling cloud data access based on reputation," *Mobile Networks and Applications*, vol. 20, pp. 828–839, 2015.

[12] D. Wenxiu, Z. Yan, and R. H. Deng, "Privacy-preserving data processing with flexible access control," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 2, pp. 363–376, 2017.

[13] W. Ding, Z. Yan, and R. H. Deng, "Encrypted data processing with homomorphic re-encryption," *Information Sciences*, vol. 409, pp. 35–55, 2017.

[14] G. Zhuo, Q. Jia, L. Guo, M. Li, and P. Li, "Privacy-preserving verifiable data aggregation and analysis for cloud-assisted mobile crowdsourcing," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, 2016, pp. 1–9.

[15] X. Yan, B. Zeng, and X. Zhang, "Privacy-preserving and customization-supported data aggregation in mobile crowdsensing," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 19 868–19 880, 2022.

[16] S. Zhao, F. Li, H. Li, R. Lu, S. Ren, H. Bao, J.-H. Lin, and S. Han, "Smart and practical privacy-preserving data aggregation for fog-based smart grids," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 521–536, 2020.

[17] H. Wu, L. Wang, and G. Xue, "Privacy-aware task allocation and data aggregation in fog-assisted spatial crowdsourcing," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 589–602, 2020.

[18] J. Zhao, H. Huang, X. Zhang, D. He, K.-K. R. Choo, and Z. L. Jiang, "Vmemda: Verifiable multi-dimensional encrypted medical data aggregation scheme for cloud-based wireless body area networks," *IEEE Internet of Things Journal*, p. 1, 2024.

[19] R. Ma, T. Feng, J. Xiong, Q. Li, and Y. Tian, "Dscpa: A dynamic subcluster privacy-preserving aggregation scheme for mobile crowdsourcing in industrial iot," *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 1880–1892, 2024.

[20] W. Lu, Z. Ren, J. Xu, and S. Chen, "Edge blockchain assisted lightweight privacy-preserving data aggregation for smart grid," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1246–1259, 2021.

[21] L. Chen, J. Zhou, Y. Chen, Z. Cao, X. Dong, and K.-K. R. Choo, "Padp: Efficient privacy-preserving data aggregation and dynamic pricing for vehicle-to-grid networks," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 7863–7873, 2020.

[22] D. Deepakraj and K. Raja, "Hybrid data aggregation algorithm for energy efficient wireless sensor networks," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, 2021, pp. 7–12.

[23] R. Ganjavi and A. R. Sharafat, "Edge-assisted public key homomorphic encryption for preserving privacy in mobile crowdsensing," *IEEE Transactions on Services Computing*, vol. 16, no. 2, pp. 1107–1117, 2022.

[24] L. Yang, M. Zhang, S. He, M. Li, and J. Zhang, "Crowd-empowered privacy-preserving data aggregation for mobile crowdsensing," in *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2018, pp. 151–160.

[25] R. Wang, S. Zhang, Z. Yang, P. Zhang, D. Wu, Y. Lu, and A. Fedotov, "Private data aggregation based on fog-assisted authentication for mobile crowd sensing," *Security and Communication Networks*, vol. 2021, pp. 1–12, 2021.

[26] C. Xu, R. Lu, H. Wang, L. Zhu, and C. Huang, "Pavs: A new privacy-preserving data aggregation scheme for vehicle sensing systems," *Sensors*, vol. 17, no. 3, p. 500, 2017.

[27] J. Fan, Q. Li, and G. Cao, "Privacy-aware and trustworthy data aggregation in mobile sensing," in *2015 IEEE Conference on Communications and Network Security (CNS)*, 2015, pp. 31–39.

[28] J. He, L. Cai, P. Cheng, J. Pan, and L. Shi, "Consensus-based data-privacy preserving data aggregation," *IEEE Transactions on Automatic Control*, vol. 64, no. 12, pp. 5222–5229, 2019.

[29] W. Tang, J. Ren, K. Deng, and Y. Zhang, "Secure data aggregation of lightweight e-healthcare iot devices with fair incentives," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8714–8726, 2019.

[30] R. Xu, J. Joshi, and P. Krishnamurthy, "An integrated privacy preserving attribute-based access control framework supporting secure deduplication," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 2, pp. 706–721, 2019.

[31] Z. Cao, B. Lang, and J. Wang, "An efficient and fine-grained access control scheme for multidimensional data aggregation in smart grid," in *2016 IEEE Trustcom/BigDataSE/ISPA*, 2016, pp. 362–369.

[32] X. Liu, R. H. Deng, K.-K. R. Choo, and J. Weng, "An efficient privacy-preserving outsourced calculation toolkit with multiple keys," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2401–2414, 2016.

[33] D. Hankerson, S. Vanstone, and A. Menezes, "Cryptographic protocols," *Guide to Elliptic Curve Cryptography*, pp. 153–204, 2004.