sensors

MDPI

Article

PAVS: A New Privacy-Preserving Data Aggregation Scheme for Vehicle Sensing Systems

Chang Xu [1,2], Rongxing Lu [3,*], Huaxiong Wang [2], Liehuang Zhu [1] and Cheng Huang [4]

[1] School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China; xuchang@bit.edu.cn (C.X.); liehuangz@bit.edu.cn (L.Z.)

[2] Division of Mathematical Sciences, School of Physical & Mathematical Sciences, Nanyang Technological University, Singapore 639798, Singapore; hxwang@ntu.edu.sg

[3] Faculty of Computer Science, University of New Brunswick, Fredericton, NB E3B 5A3, Canada

[4] Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada; c225huan@uwaterloo.ca

* Correspondence: rxlu@ieee.org; Tel.: +86-150-6451-6966

Academic Editor: Paolo Bellavista
Received: 30 December 2016; Accepted: 27 February 2017; Published: 3 March 2017

Abstract: Air pollution has become one of the most pressing environmental issues in recent years. According to a World Health Organization (WHO) report, air pollution has led to the deaths of millions of people worldwide. Accordingly, expensive and complex air-monitoring instruments have been exploited to measure air pollution. Comparatively, a vehicle sensing system (VSS), as it can be effectively used for many purposes and can bring huge financial benefits in reducing high maintenance and repair costs, has received considerable attention. However, the privacy issues of VSS including vehicles' location privacy have not been well addressed. Therefore, in this paper, we propose a new privacy-preserving data aggregation scheme, called PAVS, for VSS. Specifically, PAVS combines privacy-preserving classification and privacy-preserving statistics on both the mean $E(\cdot)$ and variance $Var(\cdot)$, which makes VSS more promising, as, with minimal privacy leakage, more vehicles are willing to participate in sensing. Detailed analysis shows that the proposed PAVS can achieve the properties of privacy preservation, data accuracy and scalability. In addition, the performance evaluations via extensive simulations also demonstrate its efficiency.

Keywords: vehicle sensing; data aggregation; privacy-preserving aggregation; privacy-preserving data statistics

1. Introduction

Air pollution has become a major environmental risk factor for ill health and death. Epidemiological studies have showed that long-term exposure to PM 2.5 can cause heart disease, stroke, and lung cancer, etc. [1]. In order to attain air pollution monitoring, a series of solutions have been proposed [2–4]. However, traditional monitoring equipment is usually stationary, complex, and expensive due to the high cost of construction and maintenance. In contrast, vehicle sensing systems (VSS) have attracted more attention, since vehicles can be equipped with various kinds of sensors that can achieve collection and concentration measurements of a range of pollutants [5]. Specifically, the sensing data are firstly collected by vehicle sensors [6], transferred to roadside units (RSUs) by vehicle wireless transmitters via vehicular ad hoc networks (VANET) [7,8], and then relayed to remote servers by RSUs.

In recent years, VSS has been regarded as a new tool to monitor gas concentration and has attracted more and more attention. Lee et al. [9] pointed out that VSS can be used to collect data when criminals spread poisonous chemicals in flight. Hu et al. [10] proposed exploiting VSS to achieve carbon dioxide monitoring. Specifically, the vehicles can be taxis or buses that collect carbon dioxide concentration and periodically report their locations and concentration. In addition, VSS can also be used for traffic monitoring. According to [11], average speed or traffic density should be collected by departments of transportation in the USA for traffic monitoring purposes. Though the traditional technologies can help collect these data, these technologies suffer high maintenance and repair costs.

Designing a VSS refers to numerous problems, e.g., how to increase the sensing coverage. Accordingly, some excellent solutions have been proposed to enhance sensing coverage and reduce detection time in vehicular sensor networks [12–16]. Moveover, a series of aggregation schemes have been proposed [17–20]. However, these aggregation schemes are only used to reduce the overhead of transmitted sensing data. Specifically, all of the aforementioned studies did not consider how to hide the real identities and location information of vehicles.

Therefore, how to achieve privacy preservation [21–26] becomes one of the most critical problems for VSS. In VSS, after the sensing data are analyzed, the statistical data e.g., the mean $E(\cdot)$ and variance $Var(\cdot)$ will probably be published in public [10]. In this case, we find there exists an attack (we call it a *sensing data link attack*), in which attackers may learn the vehicle's previous location information by linking the data collected by vehicles with the published statistical data. This kind of "*sensing data link attack*" may breach the location privacy [27] of vehicles, since location privacy of vehicles may include the drivers' living places, companies, and the amusement places to which they usually go, etc. [28–30]. Moreover, leakage of privacy is possible to produce negative effects [31]. One of the possible solutions to resist the *sensing data link attack* is to encrypt data and transmit ciphertexts to RSUs. However, it causes some problems in aggregation of encrypted data, e.g., how to classify the ciphertexts on the RSU side according to where the data are collected, and how to efficiently compute statistical data from aggregation results on the service provider side.

Aiming at the above challenges, in this paper, we propose a new privacy-preserving data aggregation scheme for VSS, called PAVS. To the best of our knowledge, it is the first work to address this "sensing data link attack" and present a privacy-preserving data aggregation scheme to compute both the mean $E(\cdot)$ and variance $Var(\cdot)$ of sensing data for VSS. Specifically, the main contributions of this paper are fourfold:

- We propose new privacy-preserving data classification and privacy-preserving aggregation algorithms, so that service providers can efficiently compute the mean $E(\cdot)$ and variance $Var(\cdot)$ from aggregation results. In addition, the proposed PAVS captures data accuracy, i.e., the $E(\cdot)$, and $Var(\cdot)$ computed from each aggregation data map to a specific area and time period.
- The proposed PAVS holds privacy-preserving property. Specifically, it can resist *sensing data link attack*. After executing PAVS, RSUs cannot get any valuable information of vehicles including vehicles' previous location information and real identities.
- The PAVS scheme achieves scalability. If a service provider holds the aggregation results of areas $Area_1, ..., Area_t$, respectively, it can further compute the statistical data of a larger area that consists of $Area_1, ..., Area_t$ by performing aggregation operations, without re-executing the whole PAVS scheme.
- To demonstrate the utility and validate the efficiency of the proposed PAVS, we theoretically analyze the performance of PAVS in terms of computational cost, communication cost and storage cost. Additionally, we develop a Java simulator to simulate the computational cost on the vehicle side, RSU side and service provider side. The experiment results show that the proposed PAVS is efficient at the three sides.

The rest of the paper is organized as follows. In Section 2, we formalize the system model, security model and identify the design goal. In Section 3, we introduce bilinear pairing, related complexity assumptions, and properties of group $\mathbb{Z}^*_{p^2}$ as preliminaries. The proposed PAVS scheme is described in Section 4, followed by the security analysis in Section 5 and the performance evaluation in Section 6. The related work is given in Section 7, and we conclude this work in Section 8.

2. Models and Design Goal

In this section, we formulate the system model, the security model and identify the design goal.

2.1. System Model

In VSS, the sensing data are collected by vehicles, transmitted to RSUs, and then transferred to the service providers [6]. In our system model, the service provider further deals with the data and publishes the results of statistical analysis in public. Our model consists of four kinds of entities: trusted authority, service provider, RSUs, and vehicles (as shown in Figure 1).

- Trusted Authority (TA): TA's duty is to manage and distribute key materials to service providers, RSUs, and vehicles in the system.
- Service Provider (SP): SP deals with each aggregation result received from an RSU and gets $\mathrm{E}(\cdot)$ and $\mathrm{Var}(\cdot)$ for each area.
- RSUs: Each RSU serves as a message aggregator role in the system. An RSU aggregates the messages sent from vehicles and forwards the aggregation results to SP. Before executing aggregation operations, RSUs will first classify the messages according to where and when the sensing data are collected.
- Vehicles: Each vehicle is equipped with sensor devices. Vehicles can then collect data in different areas and transfer messages to RSUs in batch.

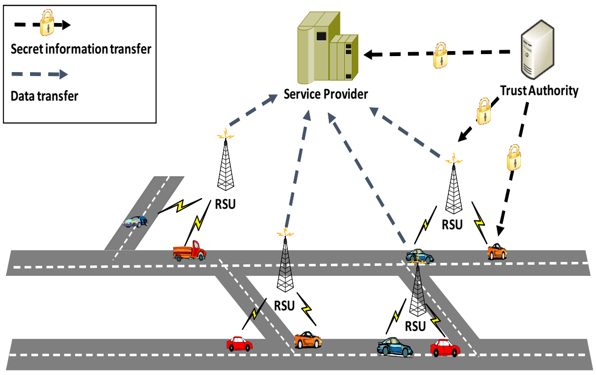

Figure 1. System model.

2.2. Security Model

In our security model, TA and SP are fully trusted. For RSUs, on one hand, RSUs will follow the designated protocol specification. On the other hand, RSUs are curious and may try to disclose vehicles' privacy information. Specifically, RSUs can get all the messages transferred in the protocol. After RSUs get all the messages, RSUs may try decrypt the ciphertext to get sensing data and launch sensing data link attacks by linking the messages sent by vehicles and the statistical results.

We will show that PAVS can resist *sensing data link attack* by introducing two levels of privacy: basic privacy and full privacy. Specifically, we will prove that PAVS holds full privacy to demonstrate that RSUs cannot link the messages sent by vehicles and the statistical results published by SP. Note that the collision of RSUs and SP is beyond the scope of this paper.

Definition 1 (Basic Privacy)**.** *When a run of the protocol is completed, RSUs cannot obtain vehicles' real identity information by communicating with vehicles.*

Definition 2 (Full Privacy)**.** *When a run of the protocol is completed, RSUs cannot obtain vehicles' real identity information and any other information with vehicles.*

2.3. Design Goal

Under the aforementioned system model and security model, our design goal is to propose an efficient privacy-preserving data aggregation scheme for VSS, so that SP can obtain more abundant information from each aggregation result without vehicles' privacy leakage. Particularly, the following four objectives should be captured:

- Privacy preservation. The privacy information of vehicles including previous location information and the real identities of vehicles should be protected.
- Accuracy. The mean E($\cdot$) and variance Var($\cdot$) computed by each aggregation result should map to a specific area and time period. Additionally, aggregation results should be generated by real RSUs, and all the sensing data should be collected by registered vehicles.
- Scalability. If SP has held the aggregation results for some small areas, E($\cdot$) and Var($\cdot$) for a larger area which consists of theses small areas should be efficiently computed without re-executing the whole scheme.
- Efficiency. The computation on the vehicle side, the RSU side and the SP side should be efficient.

3. Preliminaries

In this section, we will introduce bilinear pairing, related complexity assumptions, and properties of group $\mathbb{Z}_{p^2}^*$ that will serve as the basis of our scheme.

3.1. Bilinear Pairing and Complexity Assumptions

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative groups of order q for some large prime q, and g be a generator of $\mathbb{G}$. A bilinear map $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, which satisfies the following properties:

- Bilinearity: $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$ for all $a, b \in \mathbb{Z}_q^*$.
- Non-degeneracy: $\hat{e}(g, g) \neq 1$.
- Computability: $\hat{e}(x, y)$ can be computed efficiently.

Definition 3 (Bilinear Generator). *A bilinear parameter generator $\mathcal{G}en$ is a probability algorithm that takes a security parameter κ as input and outputs a 5-tuple $(q, g, \mathbb{G}, \mathbb{G}_T, \hat{e})$, where q is a κ-bit prime number, $(\mathbb{G}, \times)$ and $((\mathbb{G}_T, \times)$ are two groups with the same order q, $g \in \mathbb{G}$ is a generator, and $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is an admissible bilinear map.*

Definition 4 (Decisional Bilinear Diffie–Hellman (DBDH) Assumption). *Let $(q, g, \mathbb{G}, \mathbb{G}_T, \hat{e})$ be the output of the bilinear parameter generator. Given $g, g^a, g^b, g^c \in \mathbb{G}$ and $R \in \mathbb{G}_T$, where a, b, c are random elements in $\mathbb{Z}_q^*$, R is a random element in $\mathbb{G}_T$. We say an algorithm $\mathcal{B}$ that outputs $l \in \{0, 1\}$ has advantage ε in solving the DBDH problem in $\mathbb{G}$ if*

$$|\Pr[\mathcal{B}(g, g^a, g^b, g^c, \hat{e}(g, g)^{abc}) = 0] - \Pr[\mathcal{B}(g, g^a, g^b, g^c, R) = 0]| \geq \varepsilon.$$

3.2. Properties of Group $\mathbb{Z}_{p^2}^*$

Given the security parameter λ, we choose a safe prime $p = 2p' + 1$, where $|p| = \lambda$ and p' is also a prime. Then, we can calculate the Euler's totient function $\phi(p^2)$ as $\phi(p^2) = p^2(1 - 1/p) = p(p-1) = 2pp'$. That is, the order of $\mathbb{Z}_{p^2}^*$ is $2pp'$. Let $x \in \mathbb{Z}_p^*$. According to Fermat's Little Theorem, we have $x^{p-1} \equiv 1 \mod p$. Thus, for some integer k, the equality $x^{p-1} = 1 + k \cdot p$ holds. Furthermore, we obtain

$$x^{p(p-1)} = (1 + k \times p)^p = 1 + \sum_{i=1}^{p} \binom{p}{i} (k \times p)^i = 1 \mod p^2.$$

Let $y = p + 1$. When $k = 1$, we obtain

$$y^p = (p+1)^p = 1 + \sum_{i=1}^{p} \binom{p}{i} p^i = 1 \mod p^2.$$

Thus, we get the following properties of group $\mathbb{Z}_{p^2}^*$:

1 For any $x \in \mathbb{Z}_p^*$, we have $x^{p(p-1)} = 1 \mod p^2$; and
2 for any $y = p+1$, the equality $y^p = 1 \mod p^2$ holds.

4. Proposed PAVS Scheme

In this section, we present our PAVS scheme, which mainly consists of the following parts: **System Initialization**, **Data Collection** at the vehicle, **Data Aggregation** at RSU, and **Statistical Analysis** at SP.

4.1. Overview

In the **System Initialization** phase, TA will mainly execute the Parameter Generation algorithm to generate public parameters and the Key Generation algorithm to generate key materials to vehicles, RSUs and SP.

In the **Data Collection** phase, the vehicles will encrypt the sensing data by performing a Data Encryption algorithm and sign the ciphertexts by running a Message Signing algorithm. After that, the vehicles will send the messages to RSUs.

In the **Data Aggregation** phase, RSUs will classify the messages according to where and when the sensing data are collected, and aggregate the data that are collected in the same area and the same time period. Then, RSUs send the aggregation results to SP.

In the **Statistical Analysis** phase, SP will decrypt the aggregation results and get the mean E($\cdot$) and variance Var($\cdot$) for each area.

In the vehicle sensing system, the sensing data are firstly collected by vehicle sensors, transferred to RSUs by vehicle wireless transmitters via VANET, and then relayed to remote servers by RSUs. As the reviewer mentioned, the data may not be able to arrive at the data aggregation at the same time; therefore, time stamps are included in PAVS. Thus, RSUs classify the messages according to the time stamps and the area where the data are sensed. That is, only the data with the same time stamp and collected in the same area will be aggregated together. Finally, SP computes the statistic data, i.e., the E(), and Var() from each aggregation data map to a specific area and time stamp.

4.2. System Initialization

This phase is mainly comprised of the ***Parameter Generation*** algorithm, the ***Key Generation*** algorithm, and the ***List Generation*** algorithm.

Parameter Generation (PG): On input security parameter λ, TA publishes system parameters

$$(q, \mathbb{G}, \mathbb{G}_T, \hat{e}, g, p, p', \eta, H, H_1, H_2),$$

where $p = 2p' + 1$ is a safe prime, $|p| = \lambda$, p' is a large prime; $\eta \in \mathbb{Z}_p^*$ is a generator of $\mathbb{Z}_{p^2}^*$; and $(q, g, \mathbb{G}, \mathbb{G}_T, \hat{e})$ is the output of the bilinear parameter generator. $H : \{0,1\}^* \to \mathbb{Z}_q^*$, $H_1 : \{0,1\}^* \to \mathbb{G}$, and $H_2 : \{0,1\}^* \to \mathbb{Z}_p^*$ are all cryptographic hash functions.

Key Generation (KG): On input system parameters, TA generates its secret key s_0, its master private key s, area key k_0, and public parameter P_{pub}, where $s_0, s, k_0 \in \mathbb{Z}_q^*$, and $P_{pub} = g^s$. Then, the following steps are executed:

1: TA computes private key S_{L_j} for each RSU R_j, $j \in \{1, ..., \alpha\}$, where $S_{L_j} = H_1(L_j||R_j)^s$, R_j is the label of an RSU, and L_j is the location of R_j.

2: TA generates pseudo-identity PID_i for vehicle V_i, $i \in \{1, ..., \beta\}$, where $PID_i = AES_{s_0}(v_i||r_i)$, v_i is the real identity of V_i, r_i is randomly chosen in $\mathbb{Z}_q^*$, AES is the symmetric encryption algorithm, and s_0 is used to generate the symmetric encryption key.

3: After TA authenticates V_i's real identity, TA generates V_i's private key $s_i \in \mathbb{Z}_q^*$, computes V_i's public key g^{s_i} and authority key $g^{s_i r_i}$.

4: TA transfers s_i, k_0 and PID_i to V_i, S_{L_j} to R_j, and sends k_0, $\{PID_1, ..., PID_\beta\}$ and $\{g^{s_1 r_1}, ..., g^{s_\beta r_\beta}\}$ to SP.

List Generation (LG): TA generates the vehicles' public key list (as shown in Table 1, the area list (as shown in Table 2), the RSU private key list (as shown in Table 3), the random value lists (as shown in Table 4 R-value list-1 and Table 5 R-value list-2), and the vehicle authority key list (as shown in Table 6 A-key list). The vehicles' Public key list, Area list, and R-value list-2 are public, the A-key list is maintained by TA and SP secretly, and the RSU private key list and R-value list-1 are kept by TA secretly.

Table 1. Public key list.

PID	PK
PID_1	g^{s_1}
PID_2	g^{s_2}
PID_3	g^{s_3}
	...
PID_β	g^{s_β}

Table 2. Area list.

Areas
Area_1
Area_2
Area_3
...
Area_t

Table 3. Private key list.

RSUs	Private key
R_1	$H_1(L_1||R_1)^s$
R_2	$H_1(L_2||R_2)^s$
R_3	$H_1(L_3||R_3)^s$
...	...
R_α	$H_1(L_\alpha||R_\alpha)^s$

Table 4. R-value list-1.

PID	R_1
PID_1	r_1
PID_2	r_2
PID_3	r_3
...	...
PID_β	r_β

Table 5. R-value list-2.

PID	R_2
PID_1	g^{r_1}
PID_2	g^{r_2}
PID_3	g^{r_3}
...	...
PID_β	g^{r_β}

Table 6. A-key list.

PID	A- key
PID_1	$g^{s_1 r_1}$
PID_2	$g^{s_2 r_2}$
PID_3	$g^{s_3 r_2}$
...	...
PID_β	$g^{s_\beta r_\beta}$

Remark 1. *The communications between TA and each vehicle, between TA and SP, between TA and each RSU are all via private and authenticated channels. TA's secret key s_0 is used to generate vehicles' PIDs. TA uses its master private key s to generate RSUs' private keys. The area key k_0 is also known by vehicles and SP, and SP utilizes k_0 to recover the area. By using R-value list-1, TA can recover the real identity v_i according to PID_i. R-value list-2 is used by vehicles to encrypt sensing data. A-key list is utilized by SP to compute E(·) and Var(·) .*

4.3. Data Collection at Vehicle

After the vehicle V_i collects sensing data, V_i executes the following ***Data Encryption*** algorithm, ***Message Generation*** algorithm and ***Message Signing*** algorithms.

Data Encryption (DE): Assume that V_i collects $m_{i1}, ..., m_{i\theta_i}$ in $Area_1, ..., Area_{\theta_i}$, respectively, during the same time period. Let $m_{ir} \in \{0, 1, 2, \lfloor \frac{p}{\beta+1} \rfloor\}, r \in [1, \theta_i]$ $\theta_i \leq t$, where t is the number of areas and β is the number of the registered vehicles in the system. In order that SP can compute E(·) and Var(·) of the sensing data of $Area_r$, V_i encrypts m_{ir} as follows:

$$C_{ir} = (p+1)^{{m_{ir}}^2} \times \eta^{m_{ir}} \times H_2(\hat{e}(H_1(T_i, PID_i), g^{r_i})^{s_i}) \mod p^2,$$

where T_i is the time stamp.

Remark 2. *Note that the data may not be able to arrive at the data aggregation at the same time; therefore, time stamps are included in PAVS. Thus, RSUs classify the messages according to the time stamps and the area where the data are sensed. That is, only the data with the same time stamp and collected in the same area will be aggregated together. The unit of time stamp is set by TA. In real life, the unit of time stamp can be an hour or half an hour. For simplicity, the time stamp is denoted as T_i. That is, the subscript of T is denoted as i. In fact, the subscript can be set as any variable, since the time stamp is not related to the identities of the vehicles.*

Message Generation (MG): The messages sent from vehicle V_i should include the PID of V_i, so that RSUs can recover the public key from the public key list to verify the signature generated by V_i. V_i generates the message $M_i = (M_{i1}, M_{i2}, ..., M_{i\theta_i}, T_i, PID_i)$, where

$$\begin{cases} M_{i1} &= & < H_1(Area_1||T_i||k_0)^{a_{i1}}, g^{a_{i1}}, C_{i1} > \\ M_{i2} &= & < H_1(Area_2||T_i||k_0)^{a_{i2}}, g^{a_{i2}}, C_{i2} > \\ ... &= & ... \\ M_{i\theta_i} &= & < H_1(Area_{\theta_i}||T_i||k_0)^{a_{i\theta_i}}, g^{a_{i\theta_i}}, C_{i\theta_i} > . \end{cases}$$

In addition, a_{ir} is randomly chosen in $\mathbb{Z}_q^*$, $r \in [1, \theta_i]$, and $Area_r$ is the area where m_{ir} is collected. Here, M_{ir} includes two messages $H_1(Area_r||T_i||k_0)^{a_{ir}}$ and $g^{a_{ir}}$, which are used by RSUs to classify the ciphertexts.

Message Signing (MS): V_i computes the signature σ_i of M_i, where $\sigma_i = (H_1(M_i))^{s_i}$ [32]. After that, V_i sends (M_i, σ_i) to RSU R_j.

4.4. Data Aggregation at RSUs

After RSU R_j receives the messages $(M_1, \sigma_1), (M_2, \sigma_2), ..., (M_n, \sigma_n)$, R_j verifies $(\sigma_1, \sigma_2, ..., \sigma_n)$ by executing the following ***Message Verification*** algorithm. Then, R_j classifies $M_{11}, ..., M_{1\theta_1}$, $M_{21}, ..., M_{2\theta_2}$, ..., $M_{n1}, ..., M_{n\theta_n}$ according to the areas by running the ***Data Classification*** algorithm. Note that M_1= $(M_{11}, M_{12}, ..., M_{1\theta_1}, T_i, PID_i)$,..., M_n= $(M_{n1}, M_{n2}, ..., M_{n\theta_n}, T_n, PID_n)$. Finally, R_j executes the ***Data Aggregation*** algorithm.

Message Verification (MV). Firstly, R_j will check if $PID_1, ..., PID_n$ are all registered in the Public key list; for any PID_i, if it is not listed in the Public key list, R_j will not use M_i to generate aggregation results. Assume $(PID_1, PID_2, ..., PID_\delta)$ are included in the Public key list. R_j further checks if $\hat{e}(g, \sigma_i) = \hat{e}(g^{s_i}, H_1(M_i))$ holds where $i \in [1, \delta]$. If so, then M_i is a valid message. Here, we say a message is valid if it is generated by a legitimate vehicle.

Data Classification (DC): Assume $(M_1, M_2, ..., M_\delta)$ are valid messages. When R_j wants to check if $m_{i\theta_i}$ and $m_{l\theta_l}$ are collected in the same area and during the same period, where $M_{i\theta_i} =< H_1(Area_{\theta_i}||T_i||k_0)^{a_{i\theta_i}}, g^{a_{i\theta_i}}, C_{i\theta_i} >$ and $M_{l\theta_l} =< H_1(Area_{\theta_l}||T_l||k_0)^{a_{l\theta_l}}, g^{a_{l\theta_l}}, C_{l\theta_l} >$, R_j will firstly recover T_i and T_l to check if $T_i = T_l$ holds. If $T_i = T_l$ is satisfied, R_j will verify if the following equality holds

$$\hat{e}(H_1(Area_{\theta_i}||T_i||k_0)^{a_{i\theta_i}}, g^{a_{l\theta_l}}) \stackrel{?}{=} \hat{e}(H_1(Area_{\theta_l}||T_l||k_0)^{a_{l\theta_l}}, g^{a_{i\theta_i}}).$$

If so, we have $Area_{\theta_i} = Area_{\theta_l}$. That is, $m_{i\theta_i}$ and $m_{l\theta_l}$ are collected in the same area.

Data Accuracy. We can see that the scheme achieves data accuracy, since each aggregation result maps to a specific area and time period. Specifically, (1) in Data Classification phase, RSUs classify the data according to the areas where and when the data was collected, which means only data collected in the same area and during the same time period will be aggregated together; (2) the aggregation results are signed by RSUs then sent to SP. That is, the aggregation results are generated by real RSUs but not impersonated RSUs; and (3) all the sensing data are generated by registered vehicles. R_j will verify if M_i is valid by checking whether PID_i is included in the Public key list (as shown in Table 1) and verifying σ_i by using the public key corresponding to PID_i. If PID_i is not on the list or the signature is not valid, M_i will not be used any more.

Data Aggregation (DA): Assume that $(m_{1r}, m_{2r}, ..., m_{kr})$ are collected in the same area, $Area_r$, and during the same period, T_l, where $r \in \{1, ..., t\}$.

1: RSU R_j aggregates $(C_{1r}, C_{2r}, ..., C_{kr})$ by computing $\mathcal{C}_r = \prod_{i=1}^{k} C_{ir}$, and then we obtain

$$\mathcal{C}_r = (p+1)^{\sum_{i=1}^{k} {m_{ir}}^2} \times \eta^{\sum_{i=1}^{k} m_{ir}} \times \prod_{i=1}^{k} H_2(\hat{e}(H_1(PID_i, T_l), g^{s_i r_i})) \mod p^2.$$

2: Let $B_r = (L_j, PID_1, PID_2, ..., PID_k, T_l, M_{1r}, \mathcal{C}_r)$. R_j generates an ID-based signature $\sigma_r = (u_r, v_r)$ [33] on B_r by using its private key $S_{L_j} = H_1(L_j||R_j)^s$, where $u_r = H_1(L_j||R_j)^a$, a is randomly chosen from $\mathbb{Z}_q^*$, and $v_r = {S_{L_j}}^{a+H(B_r,u_r)}$. Afterwards, R_j sends (B_r, u_r, v_r) to SP.

Remark 3. *M_{1r} is included in B_r where the format of M_{1r} is $(H_1(Area_r||T_l||k_0)^{a_{1r}}, g^{a_{1r}}, C_{1r}, T_l, PID_1)$. Thus, SP can recover $Area_r$ from M_{1r} in the following Statistical Analysis phase by using $H_1(Area_r||T_l||k_0)^{a_{1r}}$ and $g^{a_{1r}}$. Since only the sensing data collected in the same area will be aggregated, SP can conclude that all sensing data are collected in$Area_r$.*

4.5. Statistical Analysis at SP

After SP receives the messages (B_r, u_r, v_r), SP will verify if B_r is valid by performing the ***Data Verification*** algorithm. If B_r is valid, SP will execute the ***Area Recovery*** algorithm to recover $Area_r$, and run the ***Data Decryption*** algorithm to decrypt $\mathcal{C}_r$ and compute E(·) and Var(·) in $Area_r$.

Data Verification (DV): After SP receives (B_r, u_r, v_r), SP verifies if the following equality holds

$$\hat{e}(g, v_r) \stackrel{?}{=} \hat{e}(P_{pub}, u_r \times H_1(L_j||R_j)^{H(B_r,u_r)}).$$

If so, (u_r, v_r) is a valid signature of B_r. SP concludes that B_r is generated by R_j.

Area Recovery (RA): SP extracts M_{1r} from B_r. According to $M_{1r} = (H_1(Area_r||T_l||k_0)^{a_{1r}}, g^{a_{1r}}, C_{1r})$, for any $Area_i, i \in \{1, ..., t\}$, SP verifies if the following equality is satisfied

$$\hat{e}(H_1(Area_r||T_l||k_0)^{a_{1r}}, g) \stackrel{?}{=} \hat{e}(H_1(Area_i||T_l||k_0), g^{a_{1r}}).$$

If, for some $Area_i$, the equality holds, then SP concludes that all the data aggregated in B_r are collected in $Area_i$.

Data Decryption (DD): SP computes statistical data E(·) and Var(·) by executing the following steps:

1: SP recovers $(g^{s_1 r_1}, g^{s_2 r_2}, ..., g^{s_k r_k})$ according to $(PID_1, PID_2, ..., PID_k)$.

2: SP computes

$$D = \frac{\mathcal{C}_r}{\prod\limits_{i=1}^{k} H_2(\hat{e}(H_1(PID_i, T_l), g^{s_i r_i}))} \mod p^2$$

and

$$\bar{D} = D^p = ((p+1)^{\sum_{i=1}^{k} {m_{ir}}^2} \times \eta^{\sum_{i=1}^{k} m_{ir}})^p = (\eta^{\sum_{i=1}^{k} m_{ir}})^p \mod p^2.$$

3: SP uses Pollard's method to recover $\sum_{i=1}^{k} m_{ir}$ and calculates

$$\hat{D} = \frac{D}{\eta^{\sum_{i=1}^{k} m_{ir}}} = (p+1)^{\sum\limits_{i=1}^{k} {m_{ir}}^2}.$$

Because m_{ir} is within a small plaintext space $\{0, 1, 2, ...k\Delta\}$, $\Sigma_{i=1}^{k}(m_{ir})^2 < p$. Therefore, we obtain

$$\begin{aligned}\hat{D} &= 1 + p \times \sum_{i=1}^{k} {m_{ir}}^2 + \sum_{i=2}^{\sum_{i=1}^{k} {m_{ir}}^2} p^i \times \binom{\sum_{i=1}^{k} {m_{ir}}^2}{i} \\ &= 1 + p \times \sum_{i=1}^{k} {m_{ir}}^2 \mod p^2.\end{aligned}$$

4: SP computes $\sum_{i=1}^{k} {m_{ir}}^2 = \frac{\hat{D}-1}{p}$. According to $\sum_{i=1}^{k} m_{ir}$ and $\sum_{i=1}^{k} {m_{ir}}^2$, SP computes $E(M)$ and $Var(M)$ of variable M for $Area_r$ where

$$E(M) = \frac{\sum_{i=1}^{k} m_{ir}}{k}$$

and

$$Var(M) = E(M^2) - (E(M))^2 = \frac{\hat{D}-1}{p \times k} - (E(M))^2.$$

Thus, SP gets $E(M)$ and $Var(M)$ of variable M for $Area_r$ at time period T_l. Similarly, SP can compute $E(M)$ and $Var(M)$ in other areas.

5. Security Analysis

Following aforementioned security requirements, our analysis will focus on how the proposed PAVS scheme can achieve the vehicles' privacy-preserving property.

Assume vehicle V_i collects sensing data in different areas, submits ciphertexts to RSU R_j, and R_j classifies and aggregates the messages and sends them to SP. We will show that the proposed PAVS scheme can resist *sensing data link attack* by showing that it achieves full privacy, which means that R_j will not get any valuable information from vehicles. In order to prove the proposed scheme achieves full privacy property, we explore the game sequence [34,35] to show that R_j cannot distinguish the messages M_i generated by vehicle V_i from random strings, where $M_i = (M_{i1}, M_{i2}, ..., M_{i\theta_i}, T_i, PID_i)$, and

$$\begin{cases} M_{i1} &= & < H_1(Area_1||T_i||k_0)^{a_{i1}}, g^{a_{i1}}, C_{i1} > \\ M_{i2} &= & < H_1(Area_2||T_i||k_0)^{a_{i2}}, g^{a_{i2}}, C_{i2} > \\ ... &= & ... \\ M_{i\theta_i} &= & < H_1(Area_{\theta_i}||T_i||k_0)^{a_{i\theta_i}}, g^{a_{i\theta_i}}, C_{i\theta_i} > . \end{cases}$$

The game sequence is explored to prove that the scheme is secure. This is because game sequence is a useful tool in taming the complexity of security proofs that might otherwise become complicated as to be nearly impossible to verify [35]. In our security proof, the attack games are played between an RSU R_j and a challenger. Both R_j and the challenger are probabilistic processes. In the proof, Game 0 and Game 1 are constructed, where Game 0 is the original attack game. If R_j cannot distinguish Game 0 and Game 1, we can conclude that it cannot distinguish the messages generated by a vehicle from random strings. The challenger generates private keys for n vehicles so that it can act as real vehicles.

Game 1. *If R_j submits (T_l, PID_l) to the challenger, the challenger will choose $(m_{l1}, ..., m_{l\theta_l})$ randomly as sensing data, answer R_j's query by normally executing the scheme, and return the messages generated by V_l to R_j.*

At some point, R_j submits (T_i, PID_i) (where T_i is not queried before. If T_i has been queried, R_j may verify if M_0^* and M_1^* are generated by real vehicles though executing *Data Classification* algorithm; however, R_j still cannot get any valuable information). The challenger generates two messages M_0^* and M_1^* to R_j, where

$$M_0^* = (< H_1(Area_1||T_i||k_0)^{a_{i1}}, g^{a_{i1}}, C_{i1} >, < H_1(Area_2||T_i||k_0)^{a_{i2}}, g^{a_{i2}}, C_{i2} >, ...,$$
$$< H_1(Area_{\theta_i}||T_i||k_0)^{a_{i\theta_i}}, g^{a_{i\theta_i}}, C_{i\theta_i} >, T_i, PID_i)$$
$$M_1^* = (< \omega_1, \omega_1', C_{i1} >, < \omega_2, \omega_2', C_{i2} >, ..., < \omega_{\theta_i}, \omega_{\theta_i}', C_{i\theta_i} >, T_i, PID_i).$$

Here, $\omega_1, \omega_1', \omega_2, \omega_2', ..., \omega_{\theta_i}, \omega_{\theta_i}'$ are all random strings, and $C_{i1}, ..., C_{i\theta_i}$ are generated by performing the scheme normally. The challenger selects a random $b \in \{0, 1\}$ uniformly, and then sends M_b^* to R_j. R_j will return 0 if it thinks that the whole message is generated by a real vehicle V_i. Otherwise, R_j returns 1. We say R_j can win Game0 with advantage $\text{Adv}_{G0}(R_j)$, where $\text{Adv}_{G0}(R_j) = |2\text{Pr}_{G0}[b = b'] - 1|$.

Game 2. *When R_j submits (T_l, PID_l) to the challenger, the challenger will choose $(m_{l1}, ..., m_{l\theta_l})$ randomly as sensing data, answer R_j's query by normally executing the scheme, and return the messages generated by V_l to R_j.*

At some time point, assume R_j queries on (T_i, PID_i) (where T_i or PID_i is not queried before). The challenger chooses two messages M_0^* and M_1^* to R_j, where

$$M_0^* = (< \beta_1, \beta_1', C_{i1} >, < \beta_2, \beta_2', C_{i2} >, ..., < \beta_{\theta_i}, \beta_{\theta_i}', C_{i\theta_i} >, T_i, PID_i),$$

$$M_1^* = (< \alpha_1, \alpha_1', \alpha_1'' >, < \alpha_2, \alpha_2', \alpha_2'' >, ..., < \alpha_{\theta_i}, \alpha_{\theta_i}', \alpha_{\theta_i}'' >, T_i, PID_i).$$

Here, $\alpha_1, ..., \alpha_{\theta_i}, \alpha_1', ..., \alpha_{\theta_i}', \alpha_1'', ..., \alpha_{\theta_i}'', \beta_1, ..., \beta_{\theta_i}$, $\beta_1', ..., \beta_{\theta_i}'$ are all random strings. The challenger selects a random $b \in \{0, 1\}$ uniformly and then sends M_b^* to R_j. R_j will return 0 if it thinks that the messages include some information generated by real vehicle V_i. Otherwise, R_j will return 1. We say R_j can win Game1 with advantage $\mathrm{Adv}_{G1}(R_j)$, where $\mathrm{Adv}_{G1}(R_j) = |2\mathrm{Pr}_{G1}[b = b'] - 1|$.

If the advantage with which R_j wins Game0 and Game1 is both negligible, we can conclude that R_j cannot get any valuable information.

We conclude that R_j cannot distinguish the message generated by registered vehicles with random strings. Firstly, the advantage $\mathrm{Adv}_{G0}(R_j)$ with which R_j wins Game 0 is negligible. That is, R_j cannot distinguish $(H_1(Area_k||T_i||k_0)^{a_{ik}}, g^{a_{ik}})$ from (ω_k, ω_k'). Let $h = g^{a_{ik}}$. Then, $H_1(Area_k||T_i||k_0)$ can be denoted as h^{b_k} for some unknown b_k. Similarly, ω_k can be denoted as $({\omega_k}'^{c_k})$ for some unknown c_k. Since h, ${\omega_k}'$, b_k, c_k are all random elements, R_j cannot distinguish (h^{b_k}, h) from $({\omega_k}'^{c_k}, \omega_k')$.

Secondly, the advantage $\mathrm{Adv}_{G1}(R_j)$ with which R_j wins Game 1 is negligible. Assume that the challenge is to break a DBDH problem instance, i.e., to distinguish c_0 and c_1 given g^x, g^y, g^z, where $c_0 = \hat{e}(g, g)^{xyz}$, $x, y, z \in \mathbb{Z}_q^*$ and c_1 is a random element in $\mathbb{G}_T$.

The challenger sets V_i's public key g^{s_i} as g^y, and g^{r_i} as g^z. In Game1, H_1 is treated as a random oracle[36]. The output of $H_1(T_i, PID_i)$ is set as g^x. Specifically, the challenger will generate $< C_{i1}, ..., C_{i\theta_i} >$ as follows:

$$< C_{i1}, ..., C_{i\theta_i} > = < (p+1)^{{m_{i1}}^2} \times \eta^{m_{i1}} \times H_2(c_0), ..., (p+1)^{{m_{i\theta_i}}^2} \times \eta^{m_{i\theta_i}} \times H_2(c_0) >,$$

$$< \alpha_1'', ..., \alpha_{\theta_i}'', > = < (p+1)^{{m_{i1}}^2} \times \eta^{m_{i1}} \times H_2(c_1), ..., (p+1)^{{m_{i\theta_i}}^2} \times \eta^{m_{i\theta_i}} \times H_2(c_1) > .$$

R_j returns a bit b' and guesses that $\mathcal{M}_{b'}$ is generated by vehicle V_i. If R_j can distinguish a valid ciphertext from a random string with a non-negligible advantage ε, then the challenger can break the DBDH assumption with non-negligible advantage.

Therefore, we can conclude that R_j cannot get any valuable information from the messages generated by registered vehicles. Thus, the proposed PAVS scheme captures full privacy, and can resist a *sensing data link attack*.

6. Performance Evaluation

In this section, we evaluate the performance of the proposed PAVS scheme in terms of computational cost, communication cost, and storage cost. In order to ease the presentation, we give the corresponding notations in Table 7.

Table 7. Notations for storage and communication cost analysis.

Notation	Definition
n_v	Number of vehicles
n_a	Number of areas
n_r	Number of RSUs
n_{dv}	Number of collected data of vehicle V
n_{cr}	Number of collected data received by RSU R
n_{ds}	Number of aggregation results received by SP
S_{id}	Bit size of pseudo identity for vehicle
S_{rsu}	Bit size of label for RSU
S_t	Bit size of time tamp
S_a	Bit size of area name
S_q	Bit size of an element in $\mathbb{Z}_q^*$
S_{p^2}	Bit size of an element in $\mathbb{Z}_{p^2}^*$

6.1. Theoretical Analysis

According to the proposed PAVS scheme, the computational cost, communication cost and storage cost at vehicle side, RSU side, and SP side will be analyzed in this section.

6.1.1. Computational Cost

In the proposed PAVS scheme, a vehicle needs to encrypt each piece of sensing data. Additionally, the vehicle will generate a signature. Since the vehicle can choose a generic signature to sign the messages, the performance of the signature is not analyzed here. For vehicle V_i to encrypt each piece of sensing data, it needs to perform a pair operation, five exponentiations, and three multiplication operations.

If RSU R_j receives n_{dr} encrypted sensing data collected from n_a different areas, it will classify the data. Assume m_c and m_d are collected in the same area. For any message m_e, in order to verify if m_e is collected in the same area with m_c and m_d, if $Area_c = Area_e$ has been verified, it is not necessary to verify if $Area_d = Area_e$ holds. Therefore, R_j will execute at most $\mathcal{O}(n_{dr}n_a)$ pair operations to achieve data classification. For k ciphertexts, R_j aggregates them by executing $(k-1)$ multiplication operations.

Assume SP receives n_{ds} aggregation results and all the sensing data are collected in n_a areas. SP will execute at most $\mathcal{O}(n_{ds}n_a)$ pair operations to recover the areas. SP executes k multiplication operations and two exponentiation operations to compute $\bar{D}$. In addition, SP needs to use Pollard's method to recover $\sum_{i=1}^{k} m_{ir}$ and performs one multiplication operation to calculate $\hat{D}$.

6.1.2. Communication Cost

In the system initialization phase, TA will send long-term secrets to vehicles, RSUs and SP. After vehicles encrypt the sensing data, they will transfer the ciphertexts and a signature to RSUs. After that, RSUs will send messages to SP. The corresponding communication cost is listed in Table 8.

Table 8. Communication cost.

Entity	Communication Cost
TA	$(n_r + n_v)S_g + 2n_vS_{id} + (2n_v + 1)S_q$
Vehicle	$S_r + n_{dr}S_{id} + S_t + 2S_\eta + 2S_q$
RSU	$n_{dr}(2S_g + S_\eta) + S_t + S_{id} + S_g$

6.1.3. Storage Cost

The storage cost is related to the phase of system initialization. The storage overhead in TA is $n_rS_g + (3 + 2n_v)S_q$+ $2n_vS_g$ + tS_a+n_vS_{id}, where $n_rS_g + (3 + 2n_v)S_q$ is the cost to store long-term secrets for vehicles, RSUs, SP and TA itself, and $2n_vS_g$ + tS_a+n_vS_{id} is the cost to store the public lists. The storage overhead at vehicle V_i is $2S_q$+S_{id}, at RSU is S_g, and at SP is $n_v(S_{id}$+$S_g)$ +S_q as shown in Table 9.

Table 9. Storage cost.

Entity	Storage Cost
TA	$n_rS_g + (3 + 2n_v)S_q$ + $2n_vS_g$ + n_aS_a + n_vS_{id}
Vehicle	$2S_q$ + S_{id}
RSU	S_{rsu} + S_g
SP	$n_v(S_{id}$ + $S_g)$ + S_q

6.2. Experimental Simulation

6.2.1. Implementation and Experimental Settings

The performance of PAVS is independent from the security parameters and the number of hash functions. Accordingly, Table 10 shows the parameter settings. The experiment is run on a test machine with Intel(R) Core(TM) I5-4200u 1.6 GHz four-core processor, 8 GB RAM, and a Windows 8 platform based on a Java Pairing-Based library [37].

Table 10. Parameter settings.

Parameter	$\|q\|$	$\|p'\|$	$\|p\|$	H	H_1	H_2
Setting	160 bit	512 bit	513 bit	160 bit	160 bit	513 bit

6.2.2. Computational Costs on the Vehicle Side

For a vehicle V_i, it needs to encrypt the sensing data, generate M_i and sign M_i. Thus, in the experiments, the computational costs of V_i are simulated by the total runtime including encryption, signature generation, and message generation algorithms on the vehicle side. On the vehicle side, the amount of sensing data n_{dv} varies from 10 to 100. The change tendency of the computational cost on the vehicle side is shown in Figure 2. We can see that the computational cost is 1.235 s if n_{dv} is 10, and 11.141 s when n_{dv} equals 100. Therefore, the algorithms for vehicles are efficient enough.

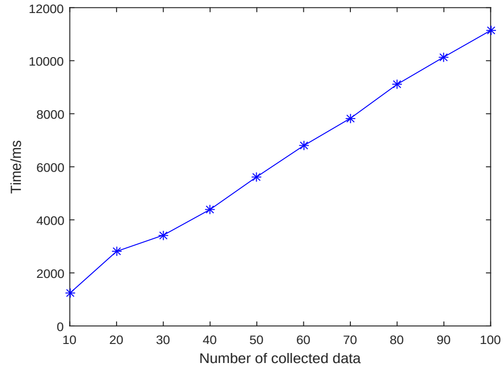

Figure 2. Computational costs on the vehicle side.

6.2.3. RSU's Computational Cost

On the RSU side, the RSUs need to verify if the received messages are valid, classify the messages and aggregate the messages. Therefore, the computational cost of an RSU is measured by the total runtime including Message Verification, Data Classification, and Data Aggregation algorithms. According to the proposed PAVS, the performance of data classification is not only related to the number of messages received by RSUs, but is also related to the number of areas which the vehicles pass by. That is, with different numbers of vehicles and areas, the computational cost of RSU will be different. Thus, we set the number of vehicles n_v as {5, 10,..., 50} and the number of areas the vehicles pass by n_a as {1, 2,..., 10}. As shown in Figure 3, although the increase of n_v and n_a leads to the increase in computational costs of RSU, the maximum running time is less than 48 s. Therefore, PAVS is efficient when computing on the RSU side, since the computation is not necessary to be in real time.

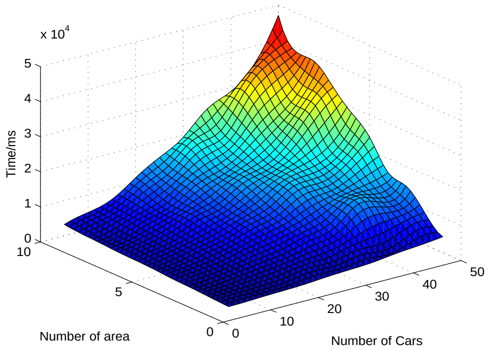

Figure 3. Computational costs on the RSU side.

6.2.4. SP's Computational Cost

On the SP side, SP will verify if B_r is valid, recover $Area_r$, and decrypt C_r. Therefore, the computational cost on the SP side is measured by the total runtime including the Data Verification algorithm, Area Recovery algorithm, and Data Decryption algorithm.

On the SP side, the number of vehicles n_v and the number of areas n_a which the vehicles pass by are still two core parameters. Accordingly, n_v is chosen from 5 to 50 and n_a is chosen from 1 to 10 to measure the computational overhead of different situations. The results are shown in Figure 4. Despite the fact that n_v and n_a increase, the running time of SP to get E(·) and Var(·) is less than 36 s, which is also acceptable.

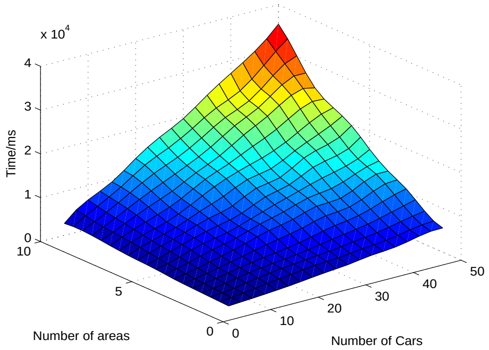

Figure 4. Computational costs on the SP side.

6.3. Scalability

Assume SP has received the aggregation results $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_n$ of $Area_1, Area_2, ..., Area_n$, respectively. If SP wants to get the statistical data E(·) and Var(·) of a larger area which includes some areas of $Area_1, Area_2, ..., Area_n$, SP can still compute the new E(·) and Var(·) without re-executing the whole scheme.

For instance, SP can get E(·) and Var(·) of a larger area which consists of $Area_1, Area_2, Area_3,$ and $Area_4$ (as shown in Figure 5), as long as SP further aggregates $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4$ and then executes Step 2, Step 3 and Step 4 of the *Data Decryption* algorithm.

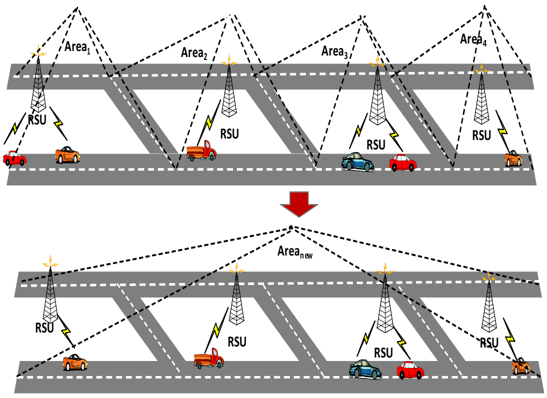

Figure 5. Area combining.

7. Related Works

In this section, we will mainly explore some of the existing work about VSS, since we propose a privacy-preserving data aggregation scheme for VSS.

In Ref. [10], Hu et al. constructed a VSS to monitor the concentration of carbon dioxide (CO_2) gas. The VSS can collect CO_2 concentration in a large field. Then, the collected data are reported to a remote server. The authors monitored the CO_2 concentration in Hsin-Chu city, Taiwan, and the data are displayed on a Google map. However, the authors did not consider security issues in their scheme.

In Ref. [38], the authors proposed deploying mobile agents to collect sensor data from some specific road segments. The mobile agent moves among vehicles and communicates with the neighbour vehicles via wireless broadcast which may not reach all the vehicles in the given segment. In order to solve the problem, they proposed an agent-based data collection scheme that can help achieve close to 100% data collection rate. Similarly, in order to enhance sensing coverage, Masutani [39] proposed a route control method. The simulation experiment shows that the sensing coverage can be enhanced significantly without increasing the number of sensing vehicles.

Different to Ref. [38,39], Zhang et al. [40] proposed the maximum coverage quality with a budget constraint problem. They proposed a new algorithm by selecting some of mobile users to maximize the coverage quality. The results of the simulation experiments showed that their algorithm achieved better performance compared with the random selection scheme.

Freschi et al. proposed a data aggregation method [41] to monitor the roughness of road surfaces. In addition, a series of data aggregation schemes [17–20] have been proposed. However, security issues are not considered in these studies. In Ref. [42], the proposed scheme achieves authentication and integrity of aggregation data by aggregating the data and the message authentication codes. In order to tolerate duplicate messages, they also presented a probabilistic data aggregation scheme. However, privacy-preservation is not considered in [42].

Wu et al. proposed a hybrid routing scheme in urban hybrid networks [43]. They firstly presented a location-based crowd sensing framework. Then, they constructed a routing switch mechanism by utilizing ad hoc solutions and RSU resources to guarantee quality of data dissemination. In Ref. [44], the authors proposed a broadcast protocol that can support dense and sparse traffic regimes.

Lee et al. [9] proposed MobEyes to support urban monitoring. For MobEyes, vehicle-local processing capabilities are utilized to extract features, and mobile agents move and collect summaries from mobile nodes. If the agents identify interest data, they will contact the involved vehicles. In Ref. [45], Lee et al. further described MobEyes. They introduced the analytic model for MobEyes performance, the effects of concurrent execution of multiple harvesting agents, the valuation network overhead, and so on. Similarly, the privacy issues are not referred to in their work.

8. Conclusions

In this paper, we have proposed PAVS—an efficient privacy-preserving data aggregation scheme for VSS. Compared with existing schemes, the proposed PAVS scheme has been identified to compute the statistical data from aggregated encryption data. To realize PAVS, we have designed concrete privacy-preserving data classification and privacy-preserving aggregation algorithms. Detailed analysis shows it can resist a *sensing data link attack* and hold data accuracy and scalability. PAVS's efficiency has been evaluated with theoretical analysis and experiments. Through extensive performance evaluations, we have demonstrated that the proposed PAVS's scheme is efficient on the SP/RSU/vehicle sides.

Acknowledgments: This work is supported by the National Natural Science Foundation of China (No. 61402037, 61272512, 61300172, 61300177) and National key research and development program 2016YFB0800301.

Author Contributions: Chang Xu, Rongxing Lu, Huaxiong Wang, Liehuang Zhu and Cheng Huang designed the scheme and the experiments, analyzed the data, and wrote the paper together.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pope, C.A., III; Burnett, R.T.; Thun, M.J.; Calle, E.E.; Krewski, D.; Ito, K.; Thurston, G.D. Lung cancer, cardiopulmonary mortality, and long-term exposure to fine particulate air pollution. *JAMA* **2002**, *287*, 1132–1141.
2. Akimoto, H. Global air quality and pollution. *Science* **2003**, *302*, 1716–1719.
3. Chung, A.; Chang, D.P.; Kleeman, M.J.; Perry, K.D.; Cahill, T.A.; Dutcher, D.; McDougall, E.M.; Stroud, K. Comparison of real-time instruments used to monitor airborne particulate matter. *J. Air Waste Manag. Assoc.* **2001**, *51*, 109–120.
4. Gupta, P.; Christopher, S.A.; Wang, J.; Gehrig, R.; Lee, Y.; Kumar, N. Satellite remote sensing of particulate matter and air quality assessment over global cities. *Atmos. Environ.* **2006**, *40*, 5880–5892.
5. Abdelhamid, S.; Hassanein, H.S.; Takahara, G. Vehicle as a mobile sensor. *Procedia Comput. Sci.* **2014**, *34*, 286–295.
6. Abdrabou, A.; Liang, B.; Zhuang, W. Delay analysis for sparse vehicular sensor networks with reliability considerations. *IEEE Trans. Wirel. Commun.* **2013**, *12*, 4402–4413.
7. Lin, X.; Lu, R. *Vehicular Ad Hoc Network Security and Privacy*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
8. Liu, Z.; Dong, M.; Zhang, B.; Ji, Y.; Tanaka, Y. RMV: Real-Time Multi-View Video Streaming in Highway Vehicle Ad-Hoc Networks (VANETs). *GLOBECOM* **2016**, 1–6, doi:10.1109/GLOCOM.2016.7842230.
9. Lee, U.; Zhou, B.; Gerla, M.; Magistretti, E.; Bellavista, P.; Corradi, A. Mobeyes: Smart mobs for urban monitoring with a vehicular sensor network. *IEEE Wirel. Commun.* **2006**, *13*, 52–57.
10. Hu, S.-C.; Wang, Y.-C.; Huang, C.-Y.; Tseng, Y.-C.; Kuo, L.-C.; Chen, C.-Y. Vehicular sensing system for CO_2 monitoring applications. In Proceedings of the IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS09), Seoul, Korea, 20–21 August 2009; pp. 168–171.
11. Rathi, A.K. A control scheme for high traffic density sectors. *Transp. Res. Part B Methodol.* **1988**, *22*, 81–101.
12. Wang, B. Coverage problems in sensor networks: A survey. *ACM Comput. Surv.* **2011**, *43*, 32 .
13. Liu, B.; Dousse, O.; Nain, P.; Towsley, D. Dynamic coverage of mobile sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **2013**, *24*, 301–311.
14. Luo, J.; Wang, D.; Zhang, Q. On the double mobility problem for water surface coverage with mobile sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **2012**, *23*, 146–159.
15. Gu, Y.; Ji, Y.; Li, J.; Zhao, B. Covering targets in sensor networks: From time domain to space domain. *IEEE Trans. Parallel Distrib. Syst.* **2012**, *23*, 1643–1656.
16. Liu, C.; Cao, G. Spatial-temporal coverage optimization in wireless sensor networks. *IEEE Trans. Mob. Comput.* **2011**, *10*, 465–478.
17. Nadeem, T.; Dashtinezhad, S.; Liao, C.; Iftode, L. Traffic view: Traffic data dissemination using car-to-car communication. *ACM SIGMOBILE Mob. Comput. Commun. Rev.* **2004**, *8*, 6–19.

18. Caliskan, M.; Graupner, D.; Mauve, M. Decentralized discovery of free parking places. In Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks, Los Angeles, CA, USA, 24–29 September 2006; pp. 30–39.
19. Lochert, C.; Scheuermann, B.; Mauve, M. Probabilistic aggregation for data dissemination in vanets. In Proceedings of the Fourth ACM International Workshop on Vehicular Ad Hoc Networks, Montreal, QC, Canada, 9–14 September 2007; pp. 1–8.
20. Dietzel, S.; Bako, B.; Schoch, E.; Kargl, F. A fuzzy logic based approach for structure-free aggregation in vehicular ad-hoc networks. In Proceedings of the Sixth ACM International Workshop on VehiculAr InterNETworking, Beijing, China, 25–29 September 2009; pp. 79–88.
21. Lu, R.; Lin, X.; Shi, Z.; Shen, X.S. A lightweight conditional privacy preservation protocol for vehicular traffic-monitoring systems. *IEEE Intell. Syst.* **2013**, *28*, 62–65.
22. Lu, R.; Lin, X.; Zhu, H.; Shen, X. An intelligent secure and privacy preserving parking scheme through vehicular communications. *IEEE Trans. Veh. Technol.* **2010**, *59*, 2772–2785.
23. Lin, X.; Lu, R.; Liang, X.; Shen, X.S. Stap: A social-tier assisted packet forwarding protocol for achieving receiver-location privacy preservation in vanets. In Proceedings of the 2011 Proceedings IEEE INFOCOM, Shanghai, China, 10–15 April 2011; pp. 2147–2155.
24. Lu, R.; Lin, X.; Liang, X.; Shen, X. A dynamic privacy-preserving key management scheme for location-based services in vanets. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 127–139.
25. Lu, R.; Lin, X.; Luan, T.H.; Liang, X.; Shen, X. Pseudonym changing at social spots: An effective strategy for location privacy in vanets. *IEEE Trans. Veh. Technol.* **2012**, *61*, 86–96.
26. Chang, S.; Zhu, H.; Dong, M.; Ota, K.; Liu, X.; Shen, X. Private and Flexible Urban Message Delivery. *IEEE Trans. Veh. Technol.* **2016**, *65*, 4900–4910.
27. Long, J.; Dong, M.; Ota, K.; Liu, A. Achieving Source Location Privacy and Network Lifetime Maximization Through Tree-Based Diversionary Routing in Wireless Sensor Networks. *IEEE Access* **2014**, *2*, 633–651.
28. Hoh, B.; Gruteser, M.; Xiong, H.; Alrabady, A. Enhancing security and privacy in traffic-monitoring systems. *IEEE Pervasive Comput.* **2006**, *5*, 38–46.
29. Liao, L.; Patterson, D.J.; Fox, D.; Kautz, H.A. Learning and inferring transportation routines. *Artif. Intell.* **2007**, *171*, 311–331.
30. Golle, P.; Partridge, K. On the anonymity of home/work location pairs. In Proceedings of the 7th International Conference Pervasive Computing, Nara, Japan, 11–14 May 2009; pp. 390–397.
31. Gruteser, M.; Grunwald, D. Anonymous usage of location-based services through spatial and temporal cloaking. In Proceedings of the First International Conference on Mobile Systems, Applications, and Services, MobiSys 2003, San Francisco, CA, USA, 5–8 May 2003.
32. Boneh, D.; Gentry, C.; Lynn, B.; Shacham, H. Aggregate and verifiably encrypted signatures from bilinear maps. In *Advances in Cryptology-EUROCRYPT 2003*; Springer: Berlin, Germany, 2003; pp. 416–432.
33. Choon, J.C.; Cheon, J.H. An identity-based signature from gap diffie-hellman groups. In *Public Key Cryptography? PKC 2003*; Springer: Berlin, Germany, 2002; pp. 18–30.
34. Shoup, V. Oaep reconsidered. In *Advances in Cryptology-CRYPTO 2001*; Springer: Berlin, Germany, 2001; pp. 239–259.
35. Shoup, V. Sequences of games: A tool for taming complexity in security proofs. *IACR Cryptol. ePrint Arch.* **2004**, *2004*, 332.
36. Bellare, M.; Rogaway, P. Random oracles are practical: A paradigm for designing efficient protocols. In Proceedings of the 1st ACM conference on Computer and Communications Security, Fairfax, VA, USA, 3–5 November 1993; pp. 62–73.
37. De Caro, A.; Iovino, V. JPBC: Java pairing based cryptography. In Proceedings of the 2011 IEEE Symposium on Computers and Communications (ISCC), Kerkyra, Corfu, Greece, 28 June–1 July 2011; pp. 850–855.
38. Huang, H.; Libman, L.; Geers, G. An agent based data collection scheme for vehicular sensor networks. In Proceedings of the 24th International Conference on Computer Communication and Networks, ICCCN 2015, Las Vegas, NV, USA, 3–6 August 2015; pp. 1–9.
39. Masutani, O. A sensing coverage analysis of a route control method for vehicular crowd sensing. In Proceeding of the 2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), St. Louis, MO, USA, 23–27 March 2015.

40. Zhang, M.; Yang, P.; Tian, C.; Tang, S.; Gao, X.; Wang, B.; Xiao, F. Quality-aware sensing coverage in budget constrained mobile crowdsensing networks. *IEEE Trans. Veh. Tech.* **2015**, *65*, 7698–7707.
41. Freschi, V.; Delpriori, S.; Klopfenstein, L.C.; Lattanzi, E.; Luchetti, G.; Bogliolo, A. Geospatial data aggregation and reduction in vehicular sensing applications: The case of road surface monitoring. In Proceedings of the ICCVE-2014, Vienna, Austria, 3–7 November 2014.
42. Du, S.; Tian, P.; Ota, K.; Zhu, H. A secure and efficient data aggregation framework in vehicular sensing networks. *Int. J. Distrib. Sens. Netw.* **2013**, *2013*, 298059.
43. Wu, D.; Zhang, Y.; Luo, J.; Li, R. Efficient data dissemination by crowdsensing in vehicular networks. In Proceedings of the 2014 IEEE 22nd International Symposium of Quality of Service (IWQoS), Hong Kong, China, 26–27 May 2014; pp. 314–319.
44. Akabane, A.T.; Villas, L.A.; Madeira, E.R.M. An adaptive solution for data dissemination under diverse road traffic conditions in urban scenarios. In Proceedings of the Wireless Communications and Networking Conference (WCNC), New Orleans, LA, USA, 9–12 March 2015; pp. 1654–1659.
45. Lee, U.; Magistretti, E.; Gerla, M.; Bellavista, P.; Corradi, A. Dissemination and harvesting of urban data using vehicular sensing platforms. *IEEE Trans. Veh. Technol.* **2009**, *58*, 882–901.